4a Edició Premi Poincaré 2007

Primer Premi

Nombres primers: anàlisi de la complexitat.

Autor: Oriol Lozano Carbasser
Centre: Aula Escola Europea (Barcelona)
Tutor: Sr. Miquel Gomila

Partint d'una situació problemàtica de tanta actualitat com és la criptografia matemàtica, per saber-ne trobar les arrels matemàtiques més profundes, analitzar-les i exposar-les amb rigor, explorant-ne amb virtuosisme les implicacions algorísmiques i computacionals, tot en un marc metodològic i heurístic remarcablement eficients.

NOMBRES PRIMERS:
Anàlisi de la Complexitat

3460629358743791518358640343380200033536377420801504613977834439628840456163670274695558067710273338903291484649576746867361500680672973256849982294175134713060467153155629165780737358063943975387837129707549222603369961280531795394931455747237797154095659698639294680127360380653926826232539927403528251263161028965803081822842452827764247160796519008797451937040525226086878573100117434233781222959427466747894565969422232074097632768922801388361131268857944271344283627004454636200325924436054507510486306322652479048427184040269481583912267530925982537961027011870036712539392460540633911524194739457545185396447282023994090709070849365614378410765696496406344741903269603648889963663737246994204425548632172469666533218712507313639708793251917881197671281119538022809944384260445948903899642907572872484700968116132219534334887670440970421855790719083757883435462538672661002637144183031216258065165336084124007083957767816382220824464482622676220393931669716917506383435300018775554674852393941078203252506454872266014618887853513040356219650523858118031569841525737930248702977790174476652276472149542457467236080726481885536504817526462173417675877031027841884390451703721233712679767872326519229728546298852758055107999799663982782342919259040484519916902662965916074408300912980827451650704621762227912005412011662136226190583572322563205886838630334920695314706679345201455235457155868076297757507673365222447863188513265824063693534338426735966871464745599314369213033816433784661931197545973914812891737073263009386011444762579866491471579638840495533883290493118358417609199248288179974376478171169435449801499543693225595051537120012542836586231599356254338556894133686449687513553351060049968327337397336165489193066805983589799281073967404636482254044010038235033240901841069580669493263398105127420431884007797631055082279895020801435295461738977941561387674459449971964118546877424106606477909182820092013047063376636595663437027502646946214122756697764317533055320447917338059676014592969681193693848147978788200377582243177441076934032633202112736510715705205176750742164117272933343509605328636776140821584006236897322664925453955694142288954982791645404803238260665873244966571021880207598194508497859666257763507071232615115268840204935893548466607645547452275906939584836406254291952428889462000997063383088763529347254845696825814805891733747597081723630017323379654759353409575141665630731288173666806091544477259579265150026669221584490458578744433996455125942074072284514345418787542384152265056858704227762738447205303033961419857179899769472732379888473080491451160793987761873805201375842224871450417073373050633407578235900850043560785073971804886170508055225080680256130579314420221849596066813493239102771311781068340666502592688764824911530613032879671256958684066722018093557010794565655227647679561697087501515281733463642593578580769487646983673021026448929065449027015299293388864503869682908265927641783109561357170771359109477910105307467954945006366966883076971709783086776607721446857800850199310375285067290903425887422719410596925072882539474833153676914250821996678140159643946116458454395192695581073662502269548848826525866810545561006080777421967373343756837125702871907434370029053100616298547489932162873113340234334659358001778189073122637851839360891507653613169025882459648293586558264981367069197045957700243704915100714003271604326499577025712201345525672024278338824890099310616119002821248537517452808049704606798423040846582069165293336669960297887078667198800211056000000069177864726194884922281982831149103588071665142389792238847852490559959838545062163630654460401526601439041278387307882782941866158918196705067312087040000000155112100433309859840479 53

La portada mostra un nombre primer de 3 799 dígits calculat en 16 hores a través d'un dels mètodes que s'analitzen en el treball. Els nombres ombrejats representen la funció zeta de Riemann.

Preface

It is always a pleasure to encounter someone who shares my fascination with the prime numbers and their surprising importance in a wide variety of practical applications, from cryptanalysis to nuclear physics. I was therefore flattered to hear from , who tells me that his project was inspired in part by my book *Prime Obsession*.

I have only been able to look over the project's Table of Contents, but it seems to me that Mr. has made a thorough survey of his topic, and presented it in a very professional way. Our knowledge of the prime numbers, of their importance and utility, is constantly expanding. It is very encouraging to see young researchers like Mr. taking up the challenges left to us by the great mathematicians of previous generations.

John Derbyshire
Author of *Prime Obsession: Berhard Riemann and the Greatest Unsolved Problem in Mathematics*
Huntington, Long Island
January 2007

Introducció

Aquesta memòria és fruit d'una recerca en el camp de la teoria dels nombres i la matemàtica computacional. L'element d'estudi i d'experimentació han estat els nombres primers i les propietats que d'aquests se'n desprenen.

La primera entrada del *Diari d'abord* -és una cronologia de les etapes- data del 3 de desembre de 2005, quan se'm va recomanar investigar en el tema (accessible en certs nivells) de la criptografia pel meu interès en la programació i la informàtica en general. En un moment inicial em vaig plantejar mostrar l'evolució històrica dels mètodes de xifra fins arribar als criptosistemes de clau pública. Foren aquests darrers els que més em sorprengueren, tan pel seu funcionament meravellosament senzill com la seguretat que proporcionaven. El seu funcionament està basat en les propietats dels nombres primers que em vaig decidir a estudiar amb deteniment per poder construir les eines necessàries a l'hora d'implementar un criptosistema assimètric, és a dir, programes generadors de primers i altres que factoritzessin el seu producte.

Tanmateix, de manera progressiva el pes dels nombres primers en el treball anà augmentant fins al punt de relegar la criptografia als apèndixs. Personalment he descobert en els nombres primers una bellesa i ordre que no esperava així que el treball s'ha orientat cap a l'anàlisi matemàtic (teòric) i computacional (pràctic) d'alguns algorismes relacionats amb els problemes fonamentals dels nombres primers: recompte, generació i factorització del seu producte. Per conèixer les propietats inherents d'aquest tipus de nombres que els feien tant especials i útils la notació O i altres conceptes matemàtics han sorgit com a eines fonamentals.

La unicitat i caràcter personal d'aquest treball no haurien d'ésser posats en dubte si es té en compte la organització, la selecció i l'apropament als temes. Utilitzant un anglicisme, certes parts d'aquest treball són un *State of the Art*, des d'un punt de vista tant rigorós com ha estat possible; en els casos en què la dificultat del tema ha superat els meus coneixements he remès el lector a l'abundant bibliografia de la qual he partit per arribar després a les conclusions en els tres àmbits senyalats, o bé m'he posat en contacte amb professors de l'esfera universitària que em poguessin ajudar a entendre els conceptes.

La principal dificultat del treball ha residit en aquesta voluntat de rigor, intentant extreure informació d'articles d'investigació matemàtica i demostrar els teoremes que avalen el funcionament de certs algorismes. El rol dels diferents tutors ha estat cabdal a l'hora d'escapçar, redirigir, sintetitzar... per tal que el treball tingués un plantejament realista i precís.

Ha estat realment el primer cop que he tingut la sensació d'estar tractant un tema amb la profunditat que mereix, aquest apropament als problemes tot i ser molt més exigent, desafiant i complex s'ha revelat enormement gratificant. He d'admetre que al acabar i durant el procés de redacció tenia proves constants de la meva suprema ignorància i limitacions matemàtiques.

Estructura interna

El treball que es presenta a continuació s'ha dividit en quatre parts, la primera presenta els conceptes més elementals de la teoria dels nombres per després poder utilitzar-los com a eines a l'hora d'analitzar algorismes i realitzar càlculs. Les altres tres parts són un reflex dels tres problemes principals dels nombres primers que ja s'han mencionat anteriorment; cada una d'aquestes tres parts té uns objectius i unes conclusions específics, si bé és cert que al final s'estableixen unes conclusions generals.

Material utilitzat

La part experimental d'aquest treball s'ha dut a terme en ordinadors, el laboratori més proper al què tinc accés. Els resultats experimentals que s'adjunten s'han obtingut en un Pentium IV a 3GHz amb una memòria RAM de 1GB que es tradueix amb una capacitat de realitzar aproximadament 10^8 operacions binàries per segon.

El codi ha estat programat en Pascal i compilat amb el *Free Pascal Compiler* 2.0.2 que té accés a tota la memòria disponible del sistema a diferència del compilador de Delphi *Turbo Pascal* 7.0 que utilitza la disponible per compatibilitat amb el sistema DOS i que per tant està molt més limitada.

Per obtenir resultats pràctics d'una banda ha estat necessari disposar de llibreries que suportessin l'aritmètica de múltiple precisió, és a dir, amb nombres de magnitud arbitrària. En una primera etapa del treball vaig programar tot un seguit de funcions per dur a terme les operacions elementals en enters però finalment em vaig decidir a utilitzar *FGInt* de Wailed Othman i *Numerix* de Michel Quercia per la seva major eficiència.

En un punt de la elaboració va esdevenir evident que necessitava un programari matemàtic potent que permetés dur a terme certs càlculs i verificar resultats. He utilitzat principalment *Mathematica* 5.2 de Wolfram Research, *Maple* 10 de MapleSoft i *Derive* 6 de Texas Instruments, aquests han estat molt útils a l'hora d' avaluar integrals impròpies, la integral logarítmica i algunes sèries infinites de convergència lenta.

Nota al lector

El treball es pot llegir a diferents nivells però cada un d'aquests permet arribar a les conclusions, si bé s'ha de senyalar que per tal de tenir una visió completa cal llegir totes les seccions i subcapítuls. Quan es parla de lectura a diferents nivells es fa referència al fet que per copsar les idees principals no cal entendre i llegir totes les demostracions dels teoremes, lemes, corol·laris i propietats que van apareixent.

Un lector molt interessat en la vessant pràctica pot simplement llegir i els objectius i conclusions de cada secció, però el que resulta més apassionant d'aquest treball no és que els algorismes funcionin sinó de quina manera i perquè ho fan. Per respondre a aquestes dues darreres qüestions resulta obvi que un anàlisi matemàtic rigorós és fonamental.

Quan s'estimi necessari adjuntar un resultat numèric no s'utilitzarà la notació americana que utilitza comes per cada tres dígits i un punt per denotar que es tracta de la part decimal ni la europea, sinó que un espai separarà els blocs de tres dígits i el punt denotarà la part real. No és una convenció arbitraria sinó que serveix per evitar ambigüitats quan es defineixen intervals.

La notació matemàtica és molt abundant i no tant sols això sinó que l'estructura interna i el desenvolupament de les seccions respon a una voluntat de formalitat matemàtica numerant els teoremes, lemes, corol·laris, propietats, algorismes i tancant les demostracions amb el símbol $\square$. S'ha partit d'un nivell de matemàtiques de I de batxillerat amb les eines del Càlcul, són les úniques que el lector hauría de necessitar perquè els altres conceptes queden definits en seccions prèvies per assegurar la bona assimilació dels continguts.

Agraïments

Voldria agrair al Sr. Fernández Millón i al Sr. l'àmplia bibliografia amb la qual em van proveir i el fet que m'introduïssin al món dels nombres primers a través la criptografia. A la Sra. per descobrir-me el món meravellós de les matemàtiques regit per l'ordre i la raó a més de donar-me unes indicacions claus per poder orientar el treball en un moment dubitatiu.

He de reconèixer l'ajuda del Sr. de la Facultat de Matemàtiques i Estadística, del Sr. de la Universitat de Girona i del Sr. de la Universitat Autònoma de Madrid a l'hora de confeccionar la distribució dels temes aquí tractats. Finalment vull agrair molt especialment al Sr. la seva paciència a l'hora d'ajudar-me a entendre certs teoremes relacionats amb la funció zeta perquè aquest treball no seria el que és sense la seva contribució.

The author wants to convey his most profound thanks to Mr. J. Derbyshire whose book and clever hints helped to shape a true understanding of the beautyful subject matter.

NOMBRES PRIMERS: Anàlisi de la Complexitat

1. Definicions prèvies 3

1.1 Teoria de la complexitat 4

1.1.1 Notació O 4

1.1.2 Complexitat d'un problema....9

1.1.3 Progrés tecnològic i algorísmic....11

1.1.4 Grans nombres: aritmètica de múltiple precisió....13

1.2 Fonaments de la Teoria dels Nombres 15

1.2.1 El Teorema Fonamental de l'Aritmètica....16

1.2.2 Quants primers hi ha?....20

1.2.3 El Teorema dels Nombres Primers....24

1.2.4 Algorisme d'Euclides....27

1.2.5 Aritmètica modular....31

2. Recompte: $\pi(x)$ i garbells 35

2.1 Garbell d'Eratosthenes....36

2.2 Garbell d'Atkin....44

2.3 Garbell de Legendre....59

2.4 Garbell de Lagarias-Miller-Odlyzko....64

2.5 Bernhard. Riemann i $\zeta(s)$....66

3. Generació i verificació 81

3.1 Tests de Monte Carlo 82

3.1.1 Test de Fermat....83

3.1.2 Test de Miller-Rabin....87

3.1.3 Test de Solovay-Strassen....99

3.1.4 Test de Lehmann....110

3.2 Tests determinístics: AKS....115

4. Factorització d'enters 118

4.1 Mètodes de complexitat exponencial 119
4.1.1 Assaig de Divisió....120
4.1.2 Mètode de Fermat....123
4.1.3 Mètode de Lehman....125
4.1.4 Mètode de Pollard....132

5. Conclusions Generals 145

6. Bibliografia 146

1. Definicions prèvies

Es presenten a continuació els conceptes fonamentals de la teoria de la complexitat i la teoria dels nombres, és un capítol imprescindible per estar en condicions d'entendre els diferents objectius que s'han proposat i les conclusions a les que s'arribarà. Aquestes dues disciplines han desenvolupat noves eines d'anàlisi des que en la dècada dels 70 es posà de relleu la seva importància trascendental en el món de la telemàtica i la computació.

1.1 Teoria de la complexitat

Oh dear Ophelia!
I am ill at these numbers.
I have not art to reckon my groans. (...)
Thine evermore most dear lady, whilst
this machine is to him,
Hamlet (Acte II, escena. 2, Línia 120)

La teoria de la complexitat és una branca de la teoria de la computació que s'ocupa de l'estudi d'algorismes: analitza els costos computacionals per resoldre un determinat problema. Els aspectes més importants són el temps d'execució i la memòria utilitzada, per aquest motiu un programador sempre intentarà optimitzar aquests recursos limitats.

1.1.1 Notació *O*

En l'anàlisi d'algorismes, si es coneix el nombre de passos necessaris per resoldre un problema, no es fan servir expressions exactes per la complexitat sinó que s'utilitzen aproximacions. La idea és molt similar al concepte matemàtic de límit, per valors grans simplificant l'expressió el valor no canvia de manera apreciable. Si sabem que un algorisme per a un problema de mida N es pot establir l'expressió $T(N)=3N^2-8N+3$, quan $\lim_{N\to\infty}$ es pot considerar que $T(N)= N^2$. Utilitzarem la notació següent:

$$T(N) \in O(N^2),$$

on la O s'utilitza per expressar la magnitud.

De la mateixa manera es pot establir comparacions entre el comportament de diversos polinomis que representen una estimació del temps d'execució en funció de l'entrada N d'un algorisme. Escriure:

$$f(N) \sim g(N)$$

representa que f té el mateix comportament assimptòtic que g quan $N \to \infty$. Per tant:

$$\lim_{N\to\infty} f(N)/g(N) = 1$$

Quan es diu

$$f(N) = O(g(N))$$

que es llegeix com, "f és O-gran de g" vol dir que existeix un enter positiu C, per totes les N, tal que:

$$|f(N)| \leq C|g(N)|$$

La notació o – petita serveix per denotar que una funció en domina clarament una altra, és a dir:

$$f(N) = o(g(N))$$

significa que

$$\lim_{N\to\infty} f(N)/g(N) = 0$$

Hi ha tres tipus de complexitat d'algorisme: polinòmic, exponencial, i logarítmic. Un algorisme és polinòmic quan el nombre d'operacions estarà acotat per la funció $T(N) = O(N^c)$, per una constant c. Un algorisme polinòmic és: constant si c val 0, lineal si c val 1, quadràtic si c val 2, cúbic si c val 3... Un algorisme és exponencial si $T(N) = O(c^{g(N)})$ per un polinomi $g(N)$ i una constant c. Si el comportament asimptòtic de l'algorisme és logarítmic, llavors el nombre d'operacions serà de $O(\ln N)^c$, on la constant $c \in (0,\infty)$.

Si es considera que un ordinador és capaç de processar 10^8 instruccions per segon; la taula següent mostra el temps d'execució per als diferents tipus d'algorismes.

Tipus	Complexitat	Operacions $N=10^8$	Temps
Pol. constant	$O(1)$	1	10^{-8} segons
Pol. lineal	$O(N)$	10^8	1 segon
Pol. quadràtic	$O(N^2)$	10^{16}	1 100 dies
Pol. Cúbic	$O(N^3)$	10^{24}	$3{,}1\cdot 10^8$ anys
Exponencial	$O(2^N)$	$10^{9\,030\,900}$	$10^{9\,030\,892}$ anys
Logarítmic	$O(\ln N)$	19	$1.9\cdot 10^{-7}$ segons

Figura 1.1

En informàtica sempre es parla d'optimitzar recursos, és a dir, temps i memòria; la solució és dissenyar algorismes més eficients. Els científics utilitzen la notació O per expressar en grans termes la complexitat d'un algorisme. Sempre s'intenten evitar algorismes amb un temps d'execució exponencial perquè per potent que sigui l'ordinador, en molts casos, no obtindríem un resultat en un temps raonable.

En articles dedicats a la computació sovint es tracten mètodes per dissenyar un algorisme de complexitat logarítmica perquè aquests són els més eficients.

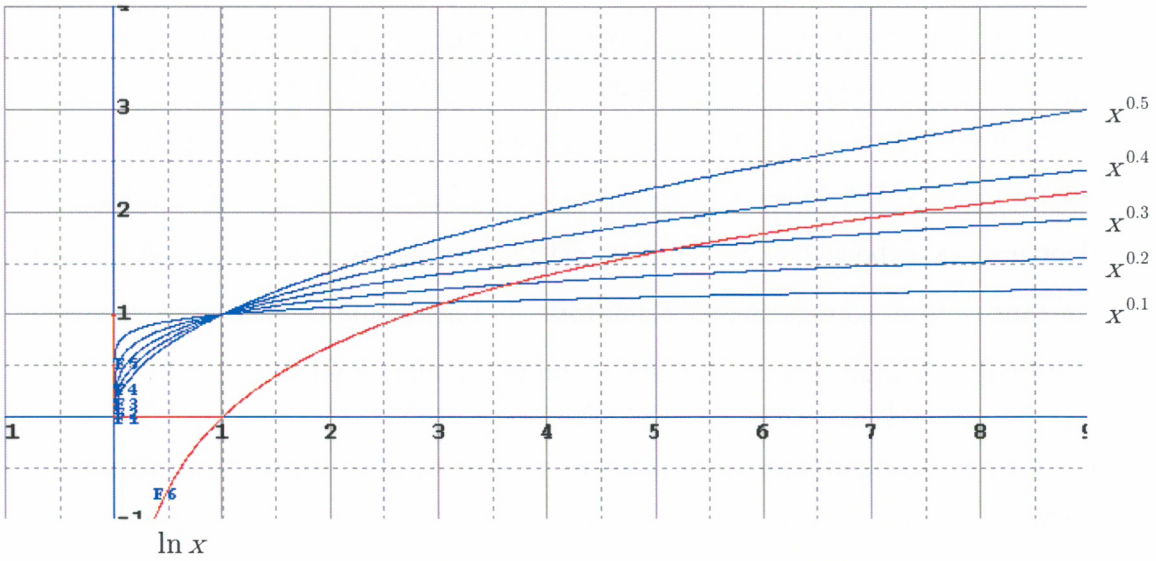


Figura 1.2

Aquest gràfic és molt revelador perquè mostra una comparació interessant entre el logaritme natural –base e- i diferents polinomis. Si l'exponent de x és major que $1/e$ les dues funcions no es toquen mai, tanmateix si és més petit no tant sols es produirà una primera intersecció entre l'1 i el 15 sinó que després per un valor de x elevat es tornen trobar. La funció logarítmica al límit, quan $x \to \infty$, sempre serà inferior que x sigui quin sigui l'exponent.

Utilitzant el mètode de Newton-Raphson, es pot aproximar que $\ln x$ intercepta $x^{0.3}$ quan x= 379.0962301, mentre que $x^{0.2}$ no ho torna a fer fins que x=332 105.1082 i $x^{0.1}$ quan x=$3.430631121 \cdot 10^{15}$.

Sembla doncs que $\ln x$ sigui x^0 tot i que no sigui cert perquè seria absurd; tanmateix a l'hora de comparar gràfics és una analogia força interessant. El logaritme neperià sempre serà inferior que x^{ε} per petita que sigui ε. Fins i tot $(\ln x)^c$ -per gran que sigui c- creix més lentament que qualsevol potència de x. Es pot il·lustrar amb un exemple el darrer raonament:

$$(\ln x)^{100} = x^{0.1}$$

$$x_{n+1} = x_n - \frac{(\ln x_n)^{100} - x_n^{0,1}}{100(\ln x_n)^{99} \cdot x_n^{-1} - 0,1x_n^{-0,9}}$$

No es creuen altra vegada fins que $x=7,94 \cdot 10^{3959}$; és una x enorme però eventualment es tornen a creuar.

i	$f_i(x)$	$x = 100$	$x = 500$	$x = 1000$	$s_i(10^{10})$	$s_i(10^{15})$
1	$c \cdot x$	100c	500c	1,000c	10^{10}/c	10^{15}/c
2	$c \cdot x^2$	10,000c	250,000c	$c \cdot 10^6$	$10^5 / \sqrt{c}$	$10^{7.5} / \sqrt{c}$
3	$c \cdot x^3$	c·10^6	$1.25c \cdot 10^8$	$c \cdot 10^9$	$2187.7 / \sqrt[3]{c}$	$10^5 / \sqrt[3]{c}$
4	$c \cdot x^4$	$c \cdot 10^8$	$6.2c \cdot 10^{10}$	$c \cdot 10^{12}$	$316.22 / \sqrt[4]{c}$	$5623.4 / \sqrt[4]{c}$
5	$c \cdot x^6$	$c \cdot 10^{12}$	$1.6c \cdot 10^{16}$	$c \cdot 10^{18}$	$46.41 / \sqrt[6]{c}$	$316.22 / \sqrt[6]{c}$
6	$c \cdot x^9$	$c \cdot 10^{18}$	$2.0c \cdot 10^{24}$	$c \cdot 10^{27}$	$13 / \sqrt[9]{c}$	$46.41 / \sqrt[9]{c}$
7	$x^{2 \ln \ln x}$	10^6	$7.3 \cdot 10^9$	$3.9 \cdot 10^{11}$	443	3,650
8	$2^{\sqrt{x}}$	1024	$5.4 \cdot 10^6$	$3.3 \cdot 10^9$	1,104	1,346
9	$2^{x/2}$	$1.12 \cdot 10^{15}$	$1.8 \cdot 10^{75}$	$3.3 \cdot 10^{150}$	60	99
10	$e^{\sqrt{x \ln 2 \ln(x \ln 2)}}$	$2.78 \cdot 10^7$	$3.56 \cdot 10^{19}$	$1.75 \cdot 10^{29}$	162	318

Taula 1.3

Aquesta taula mostra com augmenta el temps d'execució segons la complexitat; l'entrada en aquest cas s'expressa en funció del nombre de bits. Per tant es fa referència a la longitud de l'entrada i no a la magnitud del nombre; també es pot interpretar com algorismes de complexitat polinòmica analitzats segons la longitud de l'entrada.

Les columnes 3-5 mostren els temps d'execució mentre que les columnes 6 i 7 delimiten la longitud de l'entrada considerant les diferents complexitats i tenint en compte un màxim d'operacions a realitzar: 10^{10} el primer i 10^{15} el segon. Si fem una estimació assumint que un ordinador és capaç de realitzar 10^8 operacions binàries en un segon, trigarà poc més d'un minut en realitzar 10^{10} i menys de quatre mesos realitzar-ne 10^{15}.

Els algorismes de complexitat lineal o quadràtica, f_1 i f_2, poden tractar nombres exorbitantment grans en un temps raonable. Si l'algorisme realitza un nombre cúbic d'operacions poden tractar nombres de milers de dígits decimals sense problemes. Però quan l'entrada té al voltant de 15 000 dígits decimals pren 10 hores i per tant deixa de ser operatiu. Si analitzem f_4 es veu clarament que només és viable per nombres d'una longitud de pocs milers de dígits decimals; quan en té 5000 dígits decimals triga 72 dies.

Tractar un nombre de 7 dígits decimals o 23 binaris amb f_6, $f_6(\log_2 N) = (\log_2 N)^9$, necessitem 10^{12} operacions, i per tant es pot resoldre en 3 hores. Però si N té 15 o 30 dígits es necessiten 220 dies i 307 anys respectivament. Els algorismes amb complexitat de $f_7(\log_2 N) = (\log_2 N)^{2\ln\ln(\log_2 N)}$, $f_8(\log_2 N) = 2^{\sqrt{\log_2 N}}$, $f_9(\log_2 N) = 2^{(\log_2 N)/2}$ excedeixen qualsevol polinomi en la seva complexitat per una N prou gran i per tant el seu ús queda molt limitat. En criptografia la seguretat de molts criptosistemes resideix en la impossibilitat de trencar la xifra per força bruta en un temps raonable degut a la complexitat exponencial del problema de la factorització o bé del logarisme discret.

Els algorismes de complexitat polinòmica i un exponent petit poden tractar nombres grans, però quan l'exponent és superior a quatre, la longitud de l'entrada haurà de ser forçadament reduïda per ser acceptable el temps d'execució. Pels algorismes superpolinòmics o exponencials tractar nombres grans en general és inviable. És aquí on resideix la importància de la teoria de la computació, analitzar els algorismes per conèixer fins a quina entrada poden tractar en temps raonable. De fet des d'un punt de vista teòric es classifiquen els algorismes segons siguin polinòmics o no perquè tot algorismes superpolinòmic per una entrada suficientment gran excedirà un de polinòmic.

S'adjunta una taula per poder tenir referència del que representen els grans nombres. És important tenir-la en ment per reconèixer quan l'entrada per un algorisme esdevé intractable.

Analogia física	Nombre
Probabilitat de morir fulminat per un llampec(per dia)	1 entre (2^{33})
Probabilitat de guanyar el primer premi de la Primitvia	1 entre 2^{23}
Temps fins la pròxima glaciació	14000 (2^{14})anys
Temps fins que el sol s'extingeixi	$10^9(2^{30})$anys
Edat del planeta Terra	$10^9(10^{30})$anys
Edat del Univers	$10^{10}(2^{34})$anys
Nombre d'àtoms del planeta Terra	$10^{51}(2^{170})$
Nombre d'àtoms del sol	$10^{57}(2^{189})$
Nombre d'àtoms de la Via Làctea	$10^{67}(2^{233})$
Nombre d'àtoms de l'Univers (excloent matèria obscura)	$10^{77}(2^{265})$
Massa de la Terra	$5.9{\cdot}10^{24}(2^{82})$Kg.
Massa del Sol	$2{\cdot}10^{30}$ (2^{89})Kg.
Volum de la Terra	10^{21} (2^{69}) m^3
Volum de l'univers	10^{82} (2^{272}) m^3
Temps fins que la matèria radioactiva es transformi en ferro	10^{1026} anys
Temps fins que la matèria col·lapsi en forats negres	10^{1076} anys

Taula 1.4 (veure [35])

Introduirem ara la nomenclatura més generalitzada **P** i **NP** que retrobarem quan analitzem algorismes, especialment al limitar el marge d'error d'alguns d'aquests.

1.1.2 Complexitat d'un problema

Segons la teoria de complexitat, un problema es pot classificar segons l'espai i el temps mínims que requereix per ser resolt pels valors d'entrada més difícils amb en màquina de Turing. Una màquina de Turing és un model teòric per estudiar la complexitat d'un problema; té un nombre d'estats finits de memòria per controlar les operacions que du a terme però una cinta infinita de lectura i d'escriptura. Els problemes que es resolen polinòmicament en un sistema real es poden resoldre polinòmicament en una màquina de Turing i viceversa.

S'anomenen problemes tractables aquells que es poden resoldre en temps polinòmic en la màquina de Turing. La resta són problemes intractables o difícils. Quan no es pot escriure un algorisme capaç de resoldre un problema es diu que és indecidible: per exemple, un algorisme que quan se li introdueix qualsevol problema genera el codi que el soluciona.

Els problemes de decisió resolubles en un temps polinòmic respecte la mida de les dades d'entrada són de classe **P**. La classe **NP** inclou tots aquells problemes que es poden resoldre en temps polinòmic en una màquina no determinista. Si la màquina troba la solució afirmativa pot comprovar si el resultat és correcte en temps polinòmic verificant amb una informació extra anomenada certificat. La classe **co-NP** inclou tots els problemes que la resposta negativa pot ser verificada amb temps polinòmic amb un certificat. No s'explica aquí el mètode per obtenir el certificat.

Considerant les dues definicions que hem donat, es pot dir que la classe **NP** inclou la classe **P**. Tot i que hi ha problemes **NP** que demanen un temps exponencial, no s'ha pogut demostrar que **P≠NP** ni que **P=NP** ; per tant no es pot renunciar al fet que potser un dia es troben algorismes polinòmics per resoldre tots els problemes de classe **NP**, els exponencials inclosos.

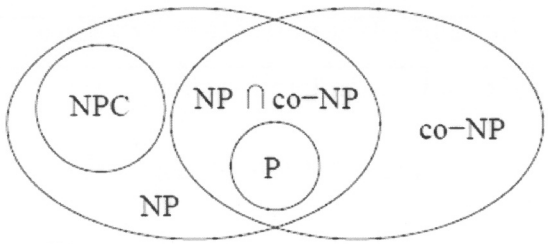


Figura 1.5

Un problema **NP**-difícil és aquell que no es pot resoldre en temps polinòmic, a no ser que **P=NP**. Un problema **NP**-complet és aquell que és **NP** i **NP**-difícil. Els problemes **NPC** es poden reduir els uns als altres, són equivalents, per tant si en resolguéssim un en temps polinòmic els hauríem resolt tots.

Hi ha una fundació sense esperit de lucre, l'Institut Clay de Matemàtiques, dedicada a extendre els coneixements matemàtics. Han ofert a qui pugui provar que **P = NP** un premi de 1.000.000$; tal descobriment representaria una revolució en la teoria de la computació. De fet només que es trobés un algorisme polinòmic per a un problema **NP**-complet, s'hauria demostrat que **P=NP**;

tanmateix molts científics es mostren escèptics perquè creuen que aquesta qüestió és independent dels axiomes inicials, i per tant és impossible de provar o invalidar.

Definició 1.1.2.1: Tipus d'algorismes

Hi ha dos grans tipus d'algorismes els deterministes s probabilístics o aleatoritzats. Els deterministes donat un mateix problema sen ɔ arriben -si hi ha d'arribar- a través del mateix camí a la mateixa solució.

Els algorismes aleatoritzats es classifiquen segons la probabilitat amb la que retornen la resposta correcta. Sigui G un algorisme aleatoritzat per al problema de decisió -ha de retornar cert o fals- D. b_1 és la probabilitat que retorni cert quan D és cert i b_2 la probabilitat que retorni cert quan D també és fals.

- G és del tipus *error nul* si $b_1=1$ i $b_2=0$
- G és del tipus *error simple* si $b_1 \geq c$ on c és una constant positiva i $b_2=0$
- G és del tipus *error doble* si $b_1 \geq \frac{1}{2}+\varepsilon$ i $b_2 \leq \frac{1}{2}-\varepsilon$

El tipus d'algorisme a error simple el es veurà força en la secció 3 perquè aquest error pot ser reduït a valors ínfims i per altra banda resulta molt eficients. Estudiant la complexitat es veu que donat un algorisme, segons la magnitud de l'entrada, aquest pot prendre temps astronòmics fora del abast fins i tot dels ordinadors més potents que existeixen.

1.1.3 Progrés tecnològic i algorísmic

El concepte d'infinit té diverses aplicacions en les matemàtiques però també té diferents matisos: no es el mateix l'infinit de nombres naturals que l'infinit dels reals. Això es deu al fet que entre 0 i 1 o entre 0 i el nombre més petit que es pugui imaginar ja hi hauria infinitud de nombres més petits.

Cramér i Dirichlet van dedicar força temps a l'estudi de l'infinit; de fet, hi ha enigmes que es basen en el concepte infinit. Si es tené un hotel d'infinites plantes i sabem que està tot ocupat, i arriben infinits clients com se'ls hi proporciona una habitació: desplaçant els infinits clients de la planta N a la $2N+1$, reservant així els pisos parells als recent arribats.

Objectivament mai es pot estar treballant amb nombres *grans*; per molts dígits que tingui un nombre sempre hi haurà un nombre finit de nombres més petits i un nombre infinit de nombres majors. En computació quan es parla de nombres grans, simplement, es fa referència a que el nombre que es tracta és major que els que es podien tractar anteriorment. Actualment estem en condicions de factoritzar nombres vuit cops majors que 30 anys enrera, i certificar la primalitat de nombres 500 cops majors.

Aquest progrés computacional es deu a dues raons: d'una banda el progrés tecnològic i de l'altra al desenvolupament de nous algorismes. Part del mèrit ha de ser evidentment atribuït a les millores de *hardware* (processadors, memòria RAM....) i a la proliferació d'ordinadors en general. Però si s'utilitzessin actualment els algorismes existents fa 30 anys molt possiblement el límit de magnitud dels nombres que es podrien factoritzar o certificar la primalitat rondaria els 50 dígits. Si la computació quàntica, basada en ordinadors amb un funcionament radicalment diferent a l'actual, acaba sent una realitat, si que es podria produir una veritable revolució.

El nombre més gran que mai ha estat provat primer (al moment de redacció del treball) és $2^{32582657}$-1, d'aproximadament 9.808.357 dígits. Si ara, al començament del segle XXI, intentem fer una estimació de les operacions dutes a terme per totes les màquines durant tota la història, és aproximadament un mol químic d'Avogadro $6,02 \cdot 10^{23} \approx 10^{24}$.

Aquesta estimació la es pot mantenir fins i tot davant la llegendària superpotència computacional dels organismes d'intel·ligència governamentals. En general quan li preguntem a algú que no té cap vinculació especial amb el món de la programació i la computació "quan diries que triga...?", la resposta més generalitzada és: "No ho sé, els ordinadors són molt ràpids: 1 segon?, 5 segons?". Com diu Knuth, tot i que la gent sovint ho oblida, els ordinadors van néixer com a instruments de càlcul; i, sí, són ràpids (afortunadament) però la qüestió és com de ràpids?

Estimacions molt optimistes dirien que 10^{12} operacions binàries –actuar sobre cada bit de memòria, canviant zeros i uns-, o una més raonable de 10^8 operacions segons l'eficiència del compilador en el processador. Per donar la idea d'una escala, per verificar la primalitat amb el mètode clàssic de buscar un divisor de N fins a $\sqrt{N}$, per un $N \approx 10^{50}$ es necessitarien aproximadament deu "mols

computacionals", deu cops tota la història dels ordinadors, per dur a terme el càlcul.

Els intents de factoritzar o provar la primalitat més complicats en general requereixen en l'ordre de 10^{18} operacions binàries. De la mateixa manera, produir una pel·lícula d'animació digital com *Buscant a Nemo* de Pixar i Disney també necessita 10^{18} operacions. És sorprenent que per dos problemes tant diferents es requereixi una milionèsima part de mol computacional atribuïble a l'espècie humana.

1.1.4 Grans nombres: aritmètica de múltiple precisió

Definició 1.1.4.1 Siguin M, N dos nombres naturals en base B

a) Sumar o restar pren $O(\log_B N + \log_B M)$ operacions binàries.

b) Multiplicar, a través de mètodes elementals, pren $O(\log_B N \cdot \log_B M)$ operacions binàries.

c) Calcular el quocient o el residu quan dividim N entre M per mètodes elementals requereix $O((\log_2 N - \log_2 M + 1)\log_2 M)$ operacions binàries

La complexitat de la suma i la resta és lineal en la longitud de l'entrada de manera que ja és prou bona. Però la multiplicació i la divisió tenen un comportament assimptòtic quadràtic a través dels mètodes elementals.

Definició 1.1.4.2 Existeixen mètodes més eficients per la multiplicació. N denota la longitud de l'entrada

a) El mètode de Karatsuba divideix cada producte en quatre de més petits i d'aquests en realitza tres; aplicant aquesta estratègia de manera recursiva assoleix $O(N^{\log_2 3}) \simeq O(N^{1.58})$.

b) Aplicant transformacions de Fourier als multiplicands es pot multiplicar en $O(N \log N \log\log N)$ operacions binàries.

Corol·lari 1.1.4.3 Es pot dividir en $O(N \log N \log\log N)$ operacions binàries.

Prova

El mètode de Newton per trobar arrels d'una funció pot ésser aplicat al problema de la divisió. El mètode iteratiu de Newton defineix les aproximacions com:

$$x_{n+1} = x_n - f(x_n)/f'(x_n)$$

la seqüència acaba convergent a la desitjada solució. La idea principal consisteix en veure que $a/b = ab^{-1}$, fet que ens evita dividir. Per obtenir el recíproc de b apliquem el mètode de Sir Isaac Newton a la funció $x^{-1} - b$.
Ara desenvolupem:

$$x_{N+1} = x_N - \frac{x_N^{-1} - b}{-x_N^{-2}} = 2x_N - bx_N^2$$

de manera que dividir és un cas particular de multiplicació. Per tant té la mateixa complexitat, comportament asimptòtic, que la multiplicació.

□

1.2 Fonaments de la Teoria dels Nombres

> La teoria del nombres ha estat vista com una de les branques de les matemàtiques pures més inútils. Contra aquesta acusació no existeix defensa vàlida; i no és mai tan apropiada com quan va dirigida contra les parts de la teoria numèrica relacionades amb els nombres primers. Una ciència es considera útil si el seu desenvolupament tendeix a accentuar les existents desigualtats en la distribució de la riquesa, o directament promou la destrucció de la vida humana. La teoria relacionada dels nombres primers no compleix aquest criteri. Aquells que la persegueixen no faran, si són savis, cap intent de justificar el seu interès en un tema tan trivial i remot, i es consolaran amb el pensament que els més grans matemàtics de tots els temps han trobat una atracció misteriosa impossible de resistir.
>
> **Godfrey Harold Hardy**
> Classe impartida el 1915 sobre els nombres primers a Cambridge

Els nombres primers sempre han captivat i fascinat als matemàtics per la seva aparent aleatorietat i per aquell component misteriós, sorprenent, exasperant, atractiu, pur, críptic, enigmàtic, impenetrable, màgic, aestètic però a la vegada de bellesa exultant. Davant d'aquest desplegament d'adjectius sembla que ens estiguem referint a quelcom de metafísic més propi de la poesia; és, per tant, paradoxal que tractem als nombres primers, pilars de les matemàtiques, la eina fonamental que ens permet ordenar el cosmos de manera lògica.

Quan tractem els nombres primers apareixen diverses qüestions estretament vinculades entre elles: existeix una funció que ens permeti calcular el $N^{è}$ nombre primer? Com reconèixer quins nombres són primers o compostos? Quina funció defineix la quantitat de nombres primers inferiors a un enter N? Existeix un nombre infinit de nombres primers?

Més enllà de la bellesa subjectiva que aquesta classe única de nombres pugui tenir, avui en dia estan de plena actualitat pel seu ús fonamental en un dels criptosistemes més utilitzats: RSA i d'altres de similar funcionament. Per aquest motiu, en les darreres dècades, des de l'aparició el 1978 de l'article de Diffie i Hellman [15], s'han fet progressos en la teoria numèrica i en l'anàlisi computacional dels nombres primers. En aquesta part del treball es defineixen els conceptes bàsics relacionats amb els nombres primers que permetran prosseguir en l'estudi del tema.

1.2.1 El Teorema Fonamental de l'Aritmètica

Definició 1.2.1.1 Un nombre natural N ¿1 és primer si té com a únics divisors positius 1 i N, és a dir, els factors trivials.

Teorema Fonamental de l'Aritmètica: Tot enter N té una factorització única com a producte de nombres primers.

$$N = p_1^{e_1} p_2^{e_2} \cdots p_r^{e_r},$$

on p és un factor primer, e la quantitat de vegades que apareix i r és el nombre de divisors diferents.

PROVA: Primer demostrarem l'existència de tal factorització. Per inducció cal discernir dos casos possibles: quan N és primer i quan és compost. Pel primer cas la factorització és trivial: 1 i N. Quan és compost, $N = a \cdot b$ on, $a, b > 1$. apliquem el mateix argument per a i a b: primer o compost. Si seguim l'estratègia de manera recurisva arribarem eventualment a la descomposició de N en factors primers.

Ara demostrarem la unicitat per reducció a l'absurd: assumim que certs enters poden ésser factoritzats en nombres primers de dues maneres diferents, ha d'haver un N mínim que compleixi aquesta propietat.

$$N = p_1 \cdots p_r = q_1 \cdots q_s$$

establim la relació $p_1 \leq \cdots \leq p_r$ i $q_1 \leq \cdots \leq q_s$ per evitar exponents. Sabem que cap $p_i = q_j$ on $1 \leq \mathrm{i} \leq r$ també $1 \leq \mathrm{j} \leq s$ perquè sinó podríem treure factor comú i N ja no seria el menor enter amb com a mínim dues factoritzacions diferents. Prenem $q_1 / p_1 = d + r / p_1$ on $0 < r < p_1 < q_1$, r no pot ser zero perquè llavors q_1 contradiria la definició de nombre primer. Rescriurem el producte anterior:

$$\begin{aligned} N = p_2 \cdots p_r &= \left(d + \tfrac{r}{p_1}\right) q_2 \cdots q_s \\ &= d q_2 \cdots q_s + \tfrac{r}{p_1} q_2 \cdots q_s \end{aligned}$$

Sigui $k = \frac{r}{p_1} q_2 \cdots q_s$ llavors $p_1 k = r q_2 \cdots q_s$ on k i r es descomponen en factors primers. L'assumpció inicial queda contradita: només pot haver una possible factorització.

□

El teorema presenta també el Problema Fonamental de l'Aritmètica, donat N trobar la seva descomposició en factors primers. A més resulta obvi perquè l'u no pot ser considerat un nombre primer: el Teorema Fonamental de l'Aritmètica –TFA- no seria cert perquè podríem afegir un nombre il·limitat de factors que no alterarien el producte, l'últim matemàtic en considerar-lo primer fou Henri Lebesgue el 1899. Aquesta "petita" convenció permet assentar la base de l'aritmètica.

Definició 1.2.1.3 $\pi(x)$ denota la quantitat de primers p tal que $p \leq x$.

Història

És una de les més importants en la teoria dels nombres. Tot i que es denota amb la mateixa lletra grega que el nombre irracional 3.14159265358979323846 que estableix una relació entre el perímetre i el radi, no té res a veure. Per aquest motiu quan es llegeix s'ha de fer molta atenció a què s'està referint: si es tracta de la funció o bé del nombre irracional i trascendental. Aquesta notació s'ha establert com notació única i va ser introduïda per Edmund Landau el 1909 en *das Handbuch* on unia dues disciplines separades, la teoria dels nombres i l'anàlisi numèric per permetre el naixement de la teoria dels nombres analítica.

Corol·lari 1.2.1.4 Si un nombre primer p divideix a $M \cdot N$, llavors p divideix a N o a M.

Prova (d'Euclides): Sigui $p_1 \cdot p_2 \cdots p_r \cdot q_1 \cdot q_2 \cdots q_s$ la descomposició factorial del producte $M \cdot N$, on $p_1 \cdots p_r$ és la factorització de N i $q_1 \cdots q_r$ llavors com que p divideix a $M \cdot N$ per força s'ha de trobar en $p_1 \cdot p_2 \cdots p_r \cdot q_1 \cdot q_2 \cdots q_s$. Llavors p divideix a N si es troba entre $p_1 \cdots p_r$ i divideix a M si és un dels $q_1 \cdots q_r$.

□

Teorema 1.2.1.6 La funció $\pi(x)$ per $x \geq 8$ queda acotada amb la següent expressió:

$$\pi(x) > \frac{\ln x}{2 \ln \ln x}$$

Prova

Es demostra la cota a través del TFA que permet aproximar

$$\pi(x) = \#\{p \leq x : p \in \mathcal{P}\}$$

Segons el TFA cada nombre té una única descomposició en factors primers; per tant el nombre de solucions a

$$\prod_{p} p_i^{a_i} \leq x$$

és igual a x perquè cada nombre tindrà la seva única factorització, on p_i denota l'i-èssim primer i a_i el nombre de vegades que apareix el factor. O dit d'una altra manera , per $x, p_i, a_i \in \mathbb{Z}$, la quantitat d'elements del conjunt

$$S = \left\{N = \prod p_i^{a_i} \leq x\right\}$$

és igual a x. Se sap del cert que $p_i^{a_i}$ no pot ser major que x per tant, es pot dir que

$$a_i \leq \left\lfloor 1 + \frac{\ln x}{\ln p_i} \right\rfloor$$

Quantes N del conjunt S es poden fabricar? Doncs per un argument combinatori, si es consideren el nombre de possibles as com a producte dels exponents de cada factor, serà com a molt:

$$x \leq \prod_{p_i \leq x} \left\lfloor 1 + \frac{\ln x}{\ln p_i} \right\rfloor$$

Per obtenir la següent aproximació cal veure que:

$$\prod_{p_i \leq x} \left\lfloor 1 + \frac{\ln x}{\ln p_i} \right\rfloor \leq \prod_{p_i \leq x} \left(1 + \frac{\ln x}{\ln p_i} \right)$$

com que tots els primers són majors o igual a 2 es pot establir que $p_i \geq 2$ i el recíproc del seu logaritme més 1 també satisfarà la condició,

$$1 + \frac{1}{\ln 2} \geq 1 + \frac{1}{\ln p_i}$$

Com que anteriorment s'ha realitzat el producte dels exponents per cada primer ara es transforma en una altra multiplicació tenint en compte quants primers es poden trobar i el valor màxim que aquest podria tenir. En conclusió:

$$x \leq \prod_{p_i \leq x} \left\lfloor 1 + \frac{\ln x}{\ln p_i} \right\rfloor \leq \left(1 + \frac{\ln x}{\ln 2}\right)^{\pi(x)}$$

Si s'aïlla l'expressió:

$$\begin{aligned}
& x \leq \left(1 + \frac{\ln x}{\ln 2}\right)^{\pi(x)} \\
& \ln x \leq \pi(x) \ln\left(1 + \frac{\ln x}{\ln 2}\right) \\
& \ln x \leq \pi(x) \ln\left(\frac{\ln x}{\ln 2}\left(1 + \frac{\ln 2}{\ln x}\right)\right) \\
& \ln x \leq \pi(x)\left[\ln\left(\frac{\ln x}{\ln 2}\right) + \ln\left(1 + \frac{\ln 2}{\ln x}\right)\right] \\
& \ln x \leq \pi(x)\left[\ln\ln x + \ln\left(\frac{1}{\ln 2}\right) + \ln\left(1 + \frac{\ln 2}{\ln x}\right)\right] \qquad * \\
& \ln x \leq \pi(x)\left[\ln\ln x - \ln\ln 2 + \ln\ln 2 + \ln\ln x\right]
\end{aligned}$$

El salt més radical es produeix al desenvolupar l'expressió *; s'ha de dir que no es tracta d'una igualtat sinó una cota o desigualtat. Per provar que el pas és correcte cal veure que

$$\ln\left(1 + \frac{\ln 2}{\ln x}\right) \leq \ln\ln 2 + \ln\ln x$$

perquè la derivada de

$$\ln\ln 2 + \ln\ln x - \ln\left(1 + \frac{\ln 2}{\ln x}\right)$$

és sempre positiva, per tant la funció és creixent i només té una intersecció al voltant de 8. De manera que la cota es compleix per $x \geq 8$

S'obté un marge molt pobre però vàlid

$$\pi(x) > \frac{\ln x}{2 \ln \ln x}$$

Resulta molt interessant d'obtenir informació sobre la distribució dels primers a partir d'un teorema tant elemental com el Teorema Fonamental de l'Aritmètica.

□

1.2.2 Quants nombres primers hi ha?

A l'època clàssica els matemàtics grecs com Pitàgores o Euclides ja coneixien el problema de la primeritat que consisteix en: donat un enter N, discernir si es tracta d'un nombre primer o no.

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103
107 109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193 197 199
211 223 227 229 233 239 241 251 257 263 269 271 277 281 283 293 307 311 313
317 331 337 347 349 353 359 367 373 379 383 389 397 401 409 419 421 431 433
439 443 449 457 461 463 467 479 487 491 499 503 509 521 523 541 547 557 563
569 571 577 587 593 599 601 607 613 617 619 631 641 643 647 653 659 661 673
677 683 691 701 709 719 727 733 739 743 751 757 761 769 773 787 797 809 811
821 823 827 829 839 853 857 859 863 877 881 883 887 907 911 919 929 937 941
947 953 967 971 977 983 991 997

Aquests són els 168 nombres primers situats en l'interval [1..1000], com es pot veure la densitat es redueix a mesura que els nombres són majors. Entre 1 i 100 hi ha 25 primers mentre que entre 900 i 1000 tant sols n'hi ha 14, en el darrer bloc de 100 nombres fins a un milió (999 999 901-1 000 000 000) n'hi ha 8. En el darrer bloc de cent nombres fins a un trilió només hi ha 4 primers; és per tant un qüestió molt legítima i apropiada preguntar-se si arriba un punt on els primers desapareixen, i eventualment quin seria aquest darrer primer.

Euclides al voltant del 300 a.C, mentre era professor a la universitat d'Alexandria, ja va demostrar que hi ha un nombre il·limitat de nombres primers (en matemàtiques sempre es pot trobar una manera alternativa per referir-se al concepte d'infinit).

Teorema 1.2.2.1 Existeixen infinits nombres primers

Prova d'Euclides (s. III a.C): Assumim que $\mathcal{P}$ és un conjunt finit amb tots els nombres primers, on p_r és el major. N serà el producte de tots els elements del conjunt més u:

$$N = 2 \cdot 3 \cdot 5 \cdots p_r + 1$$

N no és divisible per cap element del conjunt anterior, sempre deixa residu u. S'havva assumit que el conjunt comprenia tots els nombres primers, i si N no és divisible per cap primer contradiu el teorema fonamental de l'aritmètica. Per tant l'assumpció que $\mathcal{P}$ fos finit queda contradita ja que els divisors primers de N-si és compost- seran majors que p_r.

□

A partir d'aquesta demostració de la infinitud dels nombres, simplificant el producte, es pot obtenir una aproximació de $\pi(x)$ prou bonica:

$$p_{N+1} < p_1 \cdots p_N$$

tornant a aplicar aquesta idea es veu que

$$p_N p_{N+1} < (p_1 \cdots p_{n-1})(p_1 \cdots p_{n-1}) = (p_1 \cdots p_{N-1})^2$$
$$p_{N-1} \cdot p_N \cdot p_{N+1} < (p_1 \cdots p_{N-2})^{2^2}$$

Si es generalitza i s'aplica aquesta idea recursivament s'arriba a una expressió com

$$p_{N+1} < 2^{2^N}$$

tenint en compte que el primer nombre primer és 2. Ara però es canvia de notació i s'apliquen logaritmes per que es vegi més clarament què representa l'expressió.

$$\log_2 \log_2 p_{\pi(N)} < \pi(N)$$

i si es diu que a l'infinit

$$\log_2 \log_2 p_{\pi(x)} \sim \log_2 \log_2 N$$
$$\log_2 \log_2 p_N < N$$

Prova de Leonhard Euler (s. XVIII) El matemàtic suís va proposar una demostració indirecta però que és molt útil ja que està relacionada amb una funció que es retroba més endavant. La funció Zeta és una sèrie Dirichlet

$$\zeta(s) = 1 + \frac{1}{2^s} + \frac{1}{3^s} + \frac{1}{4^s} + \frac{1}{5^s} + \frac{1}{6^s} + \frac{1}{7^s} + \frac{1}{8^s} + \frac{1}{9^s} + \frac{1}{10^s} + \ldots.$$

Si es multipliquen tots dos costats de la igualtat per $\frac{1}{2^s}$ s'obté:

$$\frac{1}{2^s}\zeta(s) = \frac{1}{2^s} + \frac{1}{4^s} + \frac{1}{6^s} + \frac{1}{8^s} + \frac{1}{10^s} + \frac{1}{12^s} + \ldots$$

Al restar les dues expressions anteriors de la funció $\zeta(s)$,

$$\left(1 - \frac{1}{2^s}\right)\zeta(s) = 1 + \frac{1}{3^s} + \frac{1}{5^s} + \frac{1}{7^s} + \frac{1}{9^s} + \frac{1}{11^s} + \frac{1}{13^s} + \frac{1}{15^s} + \frac{1}{17^s} \ldots$$

El procediment ha eliminat els múltiples de dos, deixant tant sols els senars. Es repeteix el procés multiplicant per $\frac{1}{3^s}$

$$\frac{1}{3^s}\left(1 - \frac{1}{2^s}\right)\zeta(s) = \frac{1}{3^s} + \frac{1}{9^s} + \frac{1}{15^s} + \frac{1}{21^s} + \frac{1}{33^s} + \frac{1}{39^s} + \frac{1}{45^s} + \ldots$$

Si es resta aquesta darrera de l'anterior tractant la part esquerra de la igualtat com un bloc en funció de s

$$\left(1 - \frac{1}{3^s}\right)\left(1 - \frac{1}{2^s}\right)\zeta(s) = 1 + \frac{1}{5^s} + \frac{1}{7^s} + \frac{1}{11^s} + \frac{1}{13^s} + \frac{1}{17^s} + \frac{1}{19^s} + \frac{1}{23^s} + \frac{1}{25^s} + \ldots$$

Aquest cop s'han eliminat tots els múltiples de tres, curiosament aquest procés recorda al garbell d'Erastotenes que es tractarà més endavant. Ara s'hauria de repetir el procés multiplicant per 5^{-s} i restant, tot seguit amb 7^{-s}, és a dir, on la base és un nombre primer i d'aquesta manera s'aconsegueix a la part esquerra de l'expressió una llista amb primers. Si es repeteix el procés infinitament s'obté que l'expressió convergeix a 1:

$$\ldots\left(1 - \frac{1}{17^s}\right)\left(1 - \frac{1}{13^s}\right)\left(1 - \frac{1}{11^s}\right)\left(1 - \frac{1}{7^s}\right)\left(1 - \frac{1}{5^s}\right)\left(1 - \frac{1}{3^s}\right)\left(1 - \frac{1}{2^s}\right)\zeta(s) = 1$$

Si es passen els termes que multipliquen a dividir obtenir quelcom similar a

$$\zeta(s) = \frac{1}{\left(1 - \frac{1}{2^s}\right)} \times \frac{1}{\left(1 - \frac{1}{3^s}\right)} \times \frac{1}{\left(1 - \frac{1}{5^s}\right)} \times \frac{1}{\left(1 - \frac{1}{7^s}\right)} \times \frac{1}{\left(1 - \frac{1}{11^s}\right)} \ldots.$$

per tant es poden rescriure l'expressió

$$\zeta(s) = \left(1-2^{-s}\right)^{-1}\left(1-3^{-s}\right)^{-1}\left(1-5^{-s}\right)^{-1}\left(1-7^{-s}\right)^{-1}\ldots$$

$$\zeta(s) = \prod_{p}\left(1-p^{-s}\right)^{-1}$$

Per p s'entenen tots els nombres primers, és per tant un producte d'una sèrie finita o infinita: això és el que es vol demostrar. Com que la funció zeta $\zeta(s)$ es pot rescriure com una sumatòria d'una sèrie infinita s'arriba a l'expressió darrera

$$\sum_{n}\left(n^{-s}\right) = \prod_{p}\left(1-p^{-s}\right)^{-1}$$

que té nom propi: "Formula del producte de Euler". Concluïm per tant que si una banda de l'igualtat és divergent l'altra també ho és; queda provada la infinitud de nombres primers.

Quod Erat Demonstrandum

Apareix en un dels pocs papers publicats per Euler el 1737 sota el nom *Variae Observationes circa Series Infinitas* presentat a l'Acadèmia de Sant Petersburg el 25 d'abril de 1737 (veure [21]). El llatí va romandre com a llengua de cultura durant molt de temps, de fet grans científics com Sir Isaac Newton, Galileo Galilei, Leonhard Euler mateix i molts d'altres van escriure els seus llibres en llatí.

El darrer gran matemàtic a escriure en llatí probablement fou Gauss; també va ser ell dels darrers "sabis", ja que tenia coneixements amplissims de física i de totes les branques de les matemàtiques. Li costaria d'entendre que avui en dia la física i les matemàtiques siguin dues disciplines diferents i que els matemàtics s'especialitzen en només un camp com pot ser: algebra, teoria de nombres, anàlisi, teoria de funcions complexes, estadística, geometria...Tot i que la notació d'Euler és diferent en el seu paper, la idea és la mateixa.

Prova d'Ernst Kummer (1878): Sigui una seqüència finita de nombres primers: on $p_1 < p_2 < \cdots < p_r$ i sigui $N = \prod_i^r p_i > 2$. L'enter N-1 perquè no sigui primer, ha de ser un producte de primers. Té un factor primer p_i en comú amb N perquè aquest conté tots els primers.

Aplicant la propietat de divisibilitat segons la qual si $a|b$ i $a|c$ llavors $a|b-c$, s'obté que $p_i|N-(N-1)=1$ i això és absurd, per tant la proposició queda contradita i es demostra que hi ha un nombre infinit de nombres primers.

□

Prova de Thomas Stieltjes (1890): S'assumeix que hi ha un nombre finit de nombres primers. Sigui A i B dos conjunts finits en què estan continguts tots els diferents primers. Sigui P_A i P_B el producte dels elements dels respectius conjunts. Cada p_i dividirà a P_A o a P_B però no a tots dos. La suma $P_A + P_B$ no serà divisible per cap nombre dels conjunts de manera que o bé serà un altre primer o bé un compost amb primers que no pertanyen ni a A ni a B.

□

1.2.3 El Teorema dels Nombres Primers

Definició 1.2.3.1 La funció integral logarítimica es denota:

$$\mathrm{Li}(x) = \int_2^x \frac{1}{\ln t}\,dt$$

Teorema 1.2.3.2 (Chebyshev) Hi ha enters positius A i B tal que per tota $x \geq 3$,

$$\frac{Ax}{\ln x} < \pi(x) < \frac{Bx}{\ln x}$$

Chebyschev va senyalar que si algun límit existia entre $\pi(x)$ i $Cx/\ln x$, C hauria de valer forçosament 1.

Teorema dels Nombres Primers

$$\pi(x) \sim \frac{x}{\ln x}$$

Història

El TNP va ser conjecturat per Gauss als catorze anys, el 1794, i per Legendre poc després. A finals del segle XIX se'l considerava com un dels més importants i diversos matemàtics, entre ells Riemann, van dedicar grans esforços en provar la conjectura de Gauss. Es deia entre els matemàtics que aquell qui aconseguís provar el Teorema dels Nombres Primers esdevindria immortal. El 1896 dos matemàtics, de la Vallée Poussin i Hadamard, van provar el teorema de manera independent utilitzant les eines matemàtiques que havia dissenyat Riemann.

Cap dels dos no va esdevenir immortal, tanmateix van tenir una llarga vida: de la Vallé Poussin va viure fins als 96 i Hadamard fins als 98. Littlewood analitza i dona proves de l'especial longevitat d'un gran nombre de matemàtics en el llibre *The Mathematician's Art of Work* publicat el 1967 que mostra com el pensament raonat ajuda a mantenir una gran claredat mental fins a edats avançades.

De la Vallée Poussin i Hadamard van establir una definició més precisa per una constant C

$$\pi(x) = \mathrm{Li}(x) + O\left(xe^{-C\sqrt{\ln x}}\right)$$

Teorema Fonamental del Càlcul Sigui f una funció continua en [a, b], llavors la funció g definida

$$g(x) = \int_a^x f(t)\,dt \qquad a \le x \le b$$

és continua en [a, b] i derivable; a més $g'(x) = f(x)$

No donem la prova d'aquest teorema perquè no té cap especial interés en la branca de les matemàtiques que estem considerant. Si fos un treball sobre càlcul llavors molt possiblement l'afegiríem però en el nostra cas semblaria innecessari.

Corol·lari 1.2.3.5 (del TNP)

$$\mathrm{Li}(x) \sim x/\ln x$$

Prova: S'aplica la llei de l'Hôpital perquè es tracta clarament d'una indeterminació, sigui $F(x) = \mathrm{Li}(x)$ i $G(x) = x/\ln x$ quan $\lim_{x\to\infty}$ les dues funcions són divergents.

$$\lim_{x\to\infty} \frac{Li(x)}{x/\ln x} = \lim_{x\to\infty} \frac{1/\ln x}{\frac{\ln x - 1}{\ln^2 x}} = \lim_{x\to\infty} \frac{\ln^2 x}{\ln^2 x - \ln x} = \frac{\ln x}{\ln x - 1} = 1$$

de manera que la relació queda provada.

□

Història

Fins el 1914 es creia que el valor de la integral logarítimca sempre era major que el de $\pi(x)$ però Littlewood va demostrar que $\pi(x)-\mathrm{Li}(x)$ canvia de valors positius a negatius infinitat de vegades. De fet el 1933 un alumne seu, Samuel Skewe, va demostrar que si la Hipòtesi de Riemann és certa, llavors el primer creuament s'havia de produir abans de $e^{e^{e^{79}}} = 10^{10}$ bilions de trilions de trilions de dígits, major que el nombre d'àtoms de l'univers (10^{80})!

Aquest nombre era el major que havia aparegut en una demostració matemàtica fins al moment. Richard Hudson i Carter Bays l'any 2000 van fer una estimació basada en un teorema de Lehmann i ronda 1.39822×10^{316}, (veure [5]). Roger Plymen i Kuok Fai Chao del Regne Unit han publicat un article el 16 de Setembre del 2005 que demostra una violació en un subinterval de Hudson i Bays però més precís en un factor de 10, concretament en $[1.39792101\cdot10^{316}, 1.39847603\cdot10^{316}]$.

Corol·lari 1.2.3.6 La probabilitat que al prendre un valor gran de N s'obtingui un nombre primer és aproximadament $1/\ln N$

Prova

Pel TNP la quantitat de primers fins a N és $N/\ln N$ i per tant la seva densitat és $1/\ln N$. La probabilitat que al prendre un enter qualsevol en l'interval aquest sigui un primer serà igual a la densitat.

□

Corol·lari 1.2.3.7 L'eNèsim primer és $N\ln N$

Prova

Considerant un interval $[1..N]$ i Q el valor de $\pi(N)$. Aproximadament el primer nombre primer hauria d'aparèixer en N/Q, el segon a $2N/Q$, el Pè primer en $P\cdot N/Q$ i finalment el Qè primer a QN/Q, és a dir en N. Aplicant el Teorema dels Nombres Primers, el Pè primer apareixerà aproximadament en

$$\frac{P\cdot N}{Q} = \frac{P\cdot N}{N/\ln N} = P\cdot\ln N$$

Com que la majoria dels valors de P s'assemblen a N –en magnitud- es pot intercanviar i afirmar que el Nè nombre primer és $\sim N \ln N$. Per veure aquest últim raonament es posa com exemple els nombres fins a un trilió -13 dígits-: en l'interval $[1..10^{13}]$ el 90% dels nombres tenen 12 dígits o més.

$x = 10^N$	$\pi(x)$	$\pi(x) - \frac{x}{\ln x}$	$\mathrm{Li}(x) - \pi(x)$
1	4	-7.5%	55%
2	25	13.2%	20.4%
3	168	13.695%	5.95%
4	1,229	11.635%	1.38324%
5	9,592	9.445%	0.3961%
6	78,498	7.791%	0.1656%
7	664,579	6.644%	0.051%
8	5,761,455	5.775%	$1.3 \cdot 10^{-2}$%
9	50,847,534	5.098%	$3.345 \cdot 10^{-3}$%
10	455,052,511	5.561%	$6.821 \cdot 10^{-4}$%
11	4,118,054,813	4.126%	$2.81 \cdot 10^{-4}$%
12	37,607,912,018	3.767%	$1.0174 \cdot 10^{-4}$%
13	346,065,536,839	3.465%	$3.14885 \cdot 10^{-5}$%

Figura 1.6

Aquesta taula mostra com l'aproximació de l'integral logarítimica a $\pi(x)$ és molt millor que la que s'obté a partir del TNP.

1.2.4 Algorisme d'Euclides

Els algorismes d'Euclides són l'exemple a seguir en la branca de la programació basada en la teoria dels nombres per la seva gran eficiència. Ens permeten de calcular el màxim comú denominador entre dos nombres sense conèixer la seva factorització i també l'invers multiplicatiu.

Definició 1.2.4.1 Un enter N divideix un altre enter M si existeix un enter x tal que $Nx=M$. Es denota com $N \mid M$ i es diu que N és un divisor de M o que M és un múltiple de N. Ara es veuran les propietats bàsiques de divisibilitat que permetran analitzar l'algorisme del màxim comú divisor.

Propietats 1.2.4.2

a) Si $N, M \neq 0$ i $N|M$ llavors $N \leq M$

b) Si $N|M$ llavors $-N|M$ i $N|-M$

c) Si $N|M$ i $M|Z$ llavors $N|Z$

d) Si $N|M$ i $N|Z$ llavors $N|(Zv + Mu)$

Prova:

a) Per la definició 1.2.4.1 es veu clarament que $M = Nx$, per tant x¿0, $x \geq 1$ i serà un enter. Al refer la igualtat $M - N = Nx - N = N(x-1)$ i aquesta darrera igualtat és més gran que zero.

b) Per 1.2.4.1 $M = Nx$, i també $M = (-N)(-x)$ o $-M = N(-x)$

c) $M = Nx$ i $Z = My$ llavors $Z = Nxy$

d) $M = Nx$ i $Z = Ny$ llavors $Zv + Mu = Nyv + Nxu = N(vy + xu)$

Definició 1.2.4.3 Per tot enter N, $D(N)$ denota el conjunt de divisors.

Definició 1.2.4.4 S'anomena màxim comú divisor de N i M, el major enter que divideixi a N i a M, formalment és el producte dels elements de $D(N) \cap D(M)$. Cal veure que sempre hi haurà almenys un element, l'u. Si el mcd(N, M)=1 llavors es diu que N i M són coprimers. El mcd(0,0)=0.

Propietats 1.2.4.5 $x, M, N \in \mathbb{Z}$

a) mcd(N, M)= mcd(N, -M)= mcd(-N, M)= mcd(-N, -M)= $mcd(|N|,|M|)$

b) mcd(N, 0)= mcd(0, N)= mcd(N, N)=$|N|$

c) mcd(N, M)= mcd(M, N)

d) mcd(N, M)= mcd(N+ Mx, M)

e) mcd(N, M)= mcd(N mod M, M)

Prova

a) $D(N) = D(-N) = D(|N|)$ per 1.2.4.2a

b) i c) Per 1.2.4.4

d) Si M ó x són zero la propietat és immediata. Si no, cal provar que $D(N) \cap D(M) = D(N + Mx) \cap D(M)$. Sigui $N + Mx = ku$ i $M = kv$ per tant $N = ku - mx = ku - kvx = k(u - vx)$, k divideix a N.

e) mcd(N, M)= mcd(N mod M, M)=mcd($N - qM$,M). Ara s'aplica 1.2.4.5d i queda demostrat.

□

Corol·lari 1.2.4.6 Per tot $x, M, N \in \mathbb{Z}$ mcd(xN, xM)= x mcd(N, M)

Prova

S'ha vist que el mcd(N, M) és el producte dels elements $D(N) \cap D(M)$. Sigui $N' = xN$ i $M' = xM$; es proposa calcular mcd(N', M'). Es pot dividir si se sap que x és un divisor comú, per tant dividim N' i M' entre x i es busca llur màxim comú denominador. Retornatn al problema incial mcd(N', M') és igual a x mcd(N, M).

□

Algorisme 1.2.4.7 (Euclides)

```
ENTRADA: dos enters N i M
SORTIDA: retorna a
 a,b,c:Enters;
BEGIN
1 If (|N| ≥ |M|) then a=|N|; b= |M|
2                 else a=|M|; b= |N|;
3 While (b > 0) do
4    c =a; a = b; b= c mod b;
END.
```

Les línies 1 i 2 són l'aplicació de la propietat 1.2.4.5a i s'assegura que a sempre és major que b. Si no es fes, l'algorisme trobaria un cas en què mcd(0,b) i acabaria en el moment abans d'obtenir el resultat desitjat.

Les línies 3 i 4 formen un bucle i en cada iteració es canvia el divisor (b), que esdevé $a \bmod b$ el mateix moment en què a pren l'antic valor de b. Per això s'utilitza la variable intermitja c. Aquest bucle genera la seqüència:

$$(a_0, b_0), (a_1, b_1), \ldots, (a_t, b_t),$$

aquesta compleix $a_i = b_{i-1}$, $b_i = a_{i-1} \bmod b_{i-1}$ per $0 \le i \le t$. Quan i val zero, el valor ve determinat per les línies precedents 1 i 2. Per tant $(a_0, b_0) = (|a_0|, |b_0|)$.

Lema 1.2.4.8 Per cada parella (a_i, b_i) de la seqüència, es compleix

$$\mathrm{mcd}(a_i, b_i) = \mathrm{mcd}(a, b)$$

Prova:
Es es demostra la hipòtesi per inducció amb el recolzament de la propietat 1.2.4.5e. En la seqüència que hem definit es pot veure que:

$$\mathrm{mcd}(a_i, b_i) = \mathrm{mcd}(b_{i-1}, a_{i-1} \bmod b_{i-1}) = \mathrm{mcd}(a_{i-1}, b_{i-1}) = \mathrm{mcd}(N, M).$$

A més el nombre d'iteracions del bucle és finit perquè s'acaba assolint un valor $b_t = 0$. Es retorna el valor $a_t = \mathrm{mcd}(a_t, 0) = \mathrm{mcd}(N, M)$ com s'ha pogut veure.

□

Un cop s'ha demostrat que el funcionament de l'algorisme és correcte, cal veure quina és la seva complexitat; és a dir, quantes iteracions tindrà el bucle de les línies 3-4.

Teorema 1.2.4.9 Les línies 3-4 tenen la complexitat $O(\min\{\log_2 N, \log_2 M\})$.

Prova:
S'ha pogut observar anteriorment com la seqüència va descendent prou ràpidament però es pot acotar més aquesta velocitat perquè ens permetrà establir en quantes iteracions s'arriba al zero.
Hi ha dos casos possibles:

Si $b_{i+1} > \frac{1}{2} b_i = \frac{1}{2} a_{i+1}$ llavors es pot veure que $b_{i+2} = a_{i+1} - b_{i+1} < \frac{1}{2} b_i$

O bé $b_{i+1} \leq \frac{1}{2} b_i = \frac{1}{2} a_{i+1}$, és a dir, $b_{2+1} = a_{i+1} \bmod b_{i+1} < b_{i+1} \leq \frac{1}{2} b_i$

Es constata que en dues iteracions la variable divisor, b, perd un digit binari. Per tant en $2\min\{\log_2 N, \log_2 M\}$ la variable b assoleix el zero i l'algorsime s'atura. Per la definició 1.1.4.1 s'ha vist que es podia dividir a través de la representació binària del nombre, en $O((a_i - b_i + 1) b_i)$. En el nostre cas cal veure que $b_i = a_{i+1}$ per la línia 4, i s'haurà de considerar tota la seqüència on $0 \leq \mathrm{i} < t$.

Per tant, el nombre total d'operacions binàries en les línies 3-4 queda acotat per:

$$\sum_{0\le i\le t} O\big((a_i - b_i + 1)b_i\big) = \le O\left(\sum_{0\le i\le t}\big((a_i - a_{i+1} + 1)b_0 + b_i\big)\right)$$
$$= O\big(a_0 b_0 + t b_0\big) \simeq O\big(\log_2 N \cdot \log_2 M\big)$$

□

1.2.5 Aritmètica modular

Definició 1.2.5.1 Donats els enters a, b i $N \neq 0$, diu que a és congruent amb b mòdul N si $N \big| (a - b)$. La notació més establerta és $a \equiv b \pmod N$.

El cas paradigmàtic de la utilitat de l'aritmètica modular en programació és l'exponenciació modular. És útil perquè s'evita generar possibles *overflows* o sobreeixements de memòria, quan es programa o es fan càlculs amb nombres grans.

Definició 1.2.5.2 Qualsevol nombre pot ser representat de manera binària:

$$\sum_{j=0}^{k} a_j 2^j \text{ és a dir } N = a_0 + 2a_1 + 4a_2 + \ldots + 2^j a_j + \ldots + 2^k a_k$$

a_i pot ser 0 o 1 i k és el nombre de bits.

Per tant tota l'exponenciació es pot representar de la següent forma:

$$b^N = b^{a_0 + 2a_1 + 4a_2 + \ldots + 2^{k-1} a_{k-1}}$$
$$= b^{a_0} \cdot b^{2a_1} \cdot b^{4a_2} \cdots b^{2^{k-1} a_{k-1}}$$
$$= \left(b^{a_0}\right) \cdot \left(b^2\right)^{a_1} \cdot \left(b^4\right)^{a_2} \cdots \left(b^{2^{k-1}}\right)^{a_{k-1}}$$

Exemple: $a^{19} = (((a^2)^2)^2)^2 a^2 a$;

$a^{257} = (((((((a^2)^2)^2)^2)^2)^2)^2)^2 a$

Algorisme recursiu per calcular x^N segons la descomposició binària de l'exponent

```
Algorisme 1.2.5.3 exp(x,n:integer):longint;
01 BEGIN
02  if n=1 then exp:=x
03         else begin
04               if n mod 2=0 then exp=exp(x*x,n div 2)
05                            else exp:=x*exp(x*x,(n-1) div 2);
06              end;
07 END;
```

Es disposa d'un mètode per calcular a^z de manera més eficient, evitant així les z-1 multiplicacions que requeriria el mètode tradicional; la qüestió tanmateix és quina serà la complexitat del nou algorisme, és a dir, quantes multiplicacions es realitzaran. Si $b_k...b_0$ és la representació binària de l'exponent, el mètode següent tracta els bits de menys a més representatius. Es respon a la qüestió de la complexitat després de proposar una modificació de l'algorisme anterior, aquest cop en versió iterativa.

```
Algorisme 1.2.5.4 Exponenciació modular
Entrada: a,z,n;
Sortida: x;
a1,z1,x: enters
 BEGIN
1  x:=1; a1:=a; z1:=z;
2  while (z1<>0) do
3   begin
4    while (z1 mod 2=0) do
5     begin
6      z1:=z1 div 2;
7      a1:=(a1*a1)mod n;
8     end;
9    x:=(x*a1)mod n; z1:=z1-1;
10 end;
End;
```

Lema 1.2.5.5 L'algorisme 1.2.5.4 per calcular $a^z \bmod N$ té una complexitat de $O\left(2\log_2^3 z\right)$.

Prova

L'algorisme *exp'mod* processa els bits de l'exponent de menys a més significatiu, és a dir $[z_0,...z_i]$. Quan z_i=0 calcula només el quadrat -una multiplicació-, quan z_i=1 *exp'mod* multiplica i calcula un quadrat –dues multiplicacions-. Per tant per cada bit és fan dues multiplicacions a tot estirar excepte pel darrer bit, el més important, que només dóna lloc a una multiplicació. Si la llargada de bits de z és $\lceil \log_2 z \rceil$; el nombre de multiplicacions serà com a molt:

$$2\left(\lceil \log_2 z \rceil - 1\right)$$

S'ha obtingut el nombre d'operacions aritmètiques, si es té en compte la complexitat d'aquestes definida en el punt 1.1.4.1 es veu que aproximadament l'algorisme té una complexitat de $O\left(2\log_2^3 N\right)$ a través de mètodes elementals i $O\left(2\log_2^2 N\right) \approx O\left(\log_2^2 N\right)$ si s'apliquen convolucions de Fourier.

□

Per posar de relleu la importància d'aquest algorisme que apareix en repetides ocasions, es fa una estimació de càlcul si utilitzem el mètode convencional d'exponenciació. Considerem que M i e són dos enters que tenen una representació binària de 256 (2^8) bits; tot i que 2^{256} pugui semblar un nombre gran -77 dígits- no és res comparat amb el que es veurà. Al fer el càlcul de M^e es necessiten aproximadament

$$\log_2(M^e) = e \cdot \log_2(M) \approx 2^{256} \cdot 2^8 = 2^{264} \approx 10^{79}$$

bits per emmagatzemar la exponenciació. Aquest nombre és similar al volum de l'univers; és inconcebible treballar amb aquest volum de càlcul. Si es considera que hi ha 500 milions d'ordinadors amb 1 GByte de memòria. El nombre de bits disponibles per emmagatzemar és de: $500 \cdot 10^6 \cdot 10^9 \approx 10^{18}$, només suficient per emmagatzemar M^e si M i e tenen una representació de x bits.

S'ha de resoldre l'equació $10^{18} = x \cdot 2^x$; com no es pot resoldre de forma exacta s'utilitza el mètode d'aproximació iteratiu de Newton-Raphson que s'ha vist anteriorment. En el nostre cas: $f(x) = x2^x - 10^{18}$ i $f'(x) = 2^x + x2^x \ln(2)$. Després de representar gràficament la funció es pot estimar que la intersecció es produeix quan x val entre 50 i 55.

$$x_{N+1} = x_N - \frac{x2^{x_N} - 10^{18}}{2^{x_N} + x_N 2^{x_N} \ln 2}$$

Es pren x_0=50

x_1=54.30362298 (=x_0-0.696377023), x_2=54.06222680 (=x_1-0.241396174)

x_3=54.03897680 (=x_2-0.023249998), x_4=54.03878246 (=x_3-0.000194342)

x_5=54.03878244 (=x_4-0.000000020), x_6=54.03878244 (=x_5-3.6·10^{-18})

$$54.03878244 \cdot 2^{54.03878244} = 10^{18}$$

La capacitat de memòria que s'havia assumit només permetria l'exponenciació de nombres d'uns 54 dígits.

1. Conclusions

S'han revisat els conceptes elementals que seran utilitzats com a les eines que facilitar càlculs i anàlisis d'algorismes de manera que treball es pugui dirigir de manera directa cap als objectius. Aquesta part no podia ser considerada un annex perquè en el llarg i complex procés d'elaboració del treball ha estat necessari aprendre tot un seguit de conceptes matemàtics; d'entre tots aquests només apareixen en la memòria del treball aquells que són fonamentals per assolir les conclusions i assegurar que el lector segueix tots els raonaments.

Tanmateix cal remarcar que la teoria dels nombres és un mar immens on és fàcil perdre-s'hi si no es té clar cap on es va i per tant, en la mesura del possible, s'ha intentat limitar la quantitat de teoremes tot mantenint un fil conductor entre aquests. Això no descarta la possibilitat que en cada secció s'introdueixin alguns altres conceptes però calia partir d'una sòlida base per assolir les conclusions i la possibilitat de dur a terme algun tipus d'experimentació personal.

> Mai no he fet res "útil". Cap descobriment meu ha fet, ni és probable que fassi, directa o indirectament, per bé o per mal, la mínima diferència en l'amenitat del món... Jutjat per tots els estàndards pràctics, el valor de la meva vida matemàtica és nul. (...)
>
> *A Mathematician's Apology* , G. H. Hardy

2. Recompte: $\pi(x)$ i garbells

"'Com s'atrapen lleons en el desert? ' Resposta:
'En el desert hi ha molta sorra i pocs lleons; s'agafa un garbell
i es garbella la sorra, i resten els lleons. '"

Sir Arthur Stanley. Eddington (1882-1944)

Els nombres primers són les bases de l'aritmètica, les joies de la corona com deia Gauss; referint-se a la seva bellesa i ubiqüitat en la teoria dels nombres. La funció més característica d'aquesta branca de les matemàtiques és $\pi(x)$ per la qual la majoria dels grans matemàtics de tots els temps s'han sentit atrets.

Fruit d'aquest intens desig de conèixer, comprendre i comprimir diversos mètodes de càlcul per aquesta funció s'han anat succeint, cada un amb les seves característiques úniques i amb una major o menor eficiència. En aquest capítol es presenten alguns dels mètodes més coneguts per aproximar els valors de $\pi(x)$, alguns de caire més computacional o numèric i altres més matemàtics. L'apropament a aquests mètodes és tant teòric com pràctic per establir una interessant comparativa i decidir quin s'escau millor segons la magnitud de x.

2.1 El garbell d'Eratosthenes

Definició 2.1.1 Es defineix un garbell com un procés per determinar de manera molt eficient la primalitat i la factorització d'un conjunt de nombres en un interval gran de nombres. De manera indirecta ens permet calcular el valor de $\pi(x)$.

El matemàtic, geògraf i astrònom hel·lenístic grec del segle III a.C, Eratosthenes de Cyrena (actual poble de Shahhat a Líbia), va dissenyar el primer mètode per trobar tots els nombres primers fins a un enter N.

S'ha dissenyat un vector -*array*- on s'hi col·loquen tots els nombres fins a N. Tot seguit, com va optimitzar Ibn al-Banna, es prenen valors fins a $\sqrt{N}$ i es prova la divisibilitat d'aquests entre la resta d'enters de la taula. En cas que siguin divisibles s'elimina de la taula. Al acabar el procediment tan sols quedaran els nombres primers, és a dir, aquells que són tan sols divisibles entre ells i 1.

A nivell pràctic, per poder dissenyar un algorisme que, donat un enter N d'una magnitud aproximada de 10^8, calculi quins són els nombres primers menors que N, per no excedir la memòria limitada del compilador, cal inicialitzar un vector de magnitud 10^{10} però el declarat com un *array* de boolean. D'aquesta manera a cada posició, en lloc de situar un nombre que ocuparia 64 o 32 bits de memòria, tant sols s'hi emmagatzema una variable boleana, un 0 o un 1, que representen *fals* i *cert* respectivament i només ocupen un bit.

S'aconsegueix així maximitzar el límit de l'arxiu d'entrada per poder analitzar més nombres primers. Simplement es treballa amb dues variables, aquestes si de magnitud 64 bits o 10^{19}, que actuen de punters i a partir d'aquí s'executa l'algorisme. El problema del garbell d'Eratosthenes és que per enters grans pren un temps d'execució enorme, i com es veurà s'han dissenyat altres mètodes de recompte i de generació de primers molt més eficients. Degut a les limitacions de Free Pascal Compiler -FPC- els arrays no poden ser majors que 10^{11} i per tant s'han de buscar altres mètodes més eficients i moderns; tanmateix cal remarcar que l'algorisme deixa de ser pràctic per "mèrits pròpis", és a dir, pel temps d'execució que pren.

La representació gràfica mostra com el temps a partir d'un valor comença a créixer excessivament.

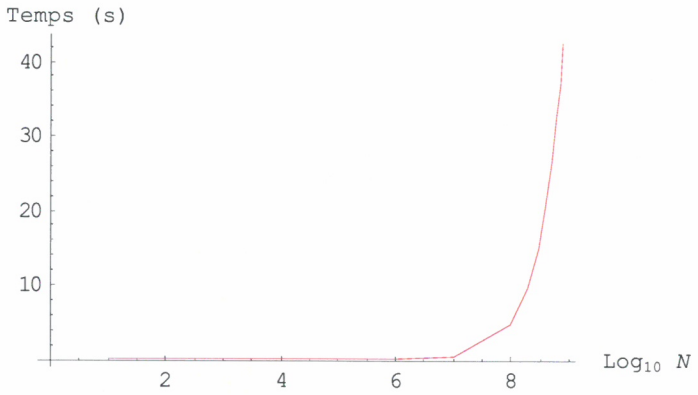


Figura 2.1

La taula de resultats ha estat completada a través de l'execució del programa per diferents valors de N en un rang de 1 a 8 inclosos. Com es demostrarà tot seguit, anar més enllà no resulta eficient.

N	$\pi(N)$	**Temps**
10^1	4	0:0:0:01
10^2	25	0:0:0:01
10^3	168	0:0:0:01
10^4	1 229	0:0:0:01
10^5	9 592	0:0:0:01
10^6	78 498	0:0:0:03
10^7	664 576	0:0:0:40
10^8	5 761 455	0:0:4:93
$2*10^8$	11 078 937	0:0:9:71
$3*10^8$	16 252 325	0:0:14:87
$4*10^8$	21 336 326	0:0:20:45
$5*10^8$	26 355 867	0:0:26:57
$6*10^8$	31 324 703	0:0:32:23
$7*10^8$	36 252 931	0:0:37:10
$8*10^8$	41 146 179	0:0:42:46
$9*10^8$	46 009 215	4:2:21:14

Taula 2.2

```
Algorisme 2.1.2
Function eratost(n:int64):int64;
var t:array[1..MaX]of boolean;
    i,j,p:int64;
BEGIN
p:=n-1;
repeat
  begin
   if t[j]=false then
    begin
     i:=j*j;
     while i<=n do
      begin
       if t[i]=false then begin
                           t[i]:=true; dec(p,1);
                          end;
       i:=i+j;
      end;
    end;
   inc(j);
  end until(j*j>=n);

 eratost:=p;
END.
```

Per calcular la complexitat de l'algorisme cal veure que en el primer *repeat*, on la variable de treball és j, es produeixen $\sqrt{N}$ iteracions. En cas que j sigui compost el temps serà constant, si és primer llavors l'algorisme entre en un altre bucle per eliminar els múltiples de la llista tot i que ja hagin estat marcats anteriorment. Cal doncs determinar quin és el nombre d'iteracions del *while*, on la variable de treball és i: per un primer $p \le \sqrt{N}$ no pot haver més que N/p múltiples tals que $r \cdot p \le N$. Retornant al *while* anterior es pot limitar el nombre d'iteracions on es marca un nombre com compost:

$$\sum_{p \le \sqrt{N}} \frac{N}{p} \le N \cdot \sum_{p \le \sqrt{N}} \frac{1}{p}$$

Teorema 2.1.3 L'expressió següent es compleix:

$$\sum_{p \le \sqrt{N}} \frac{1}{p} < 1 + \ln \sqrt{N} = 1 + \tfrac{1}{2} \ln N .$$

Prova

Per establir el límit superior ens cal en realitat provar que si $N_H = \sum_{i=1}^{N} \frac{1}{i}$, la següent expressió es complirà per N major o igual a 2:

$$\ln N < N_H < 1 + \ln N$$

és ràpidament verificable que

$$\int_i^{i+1} \frac{1}{x} dx < \frac{1}{i}, \text{ per } i \geq 1, \text{ també } \int_{i-1}^{i} \frac{1}{x} dx > \frac{1}{i}, \text{ per } i \geq 2.$$

Al sumar la primera igualtat per $1 \leq i < N$ s'obté:

$$\ln N = \int_1^N \frac{dx}{x} = \sum_{1 \leq i < N} \int_i^{i+1} \frac{dx}{x} < \sum_{1 \leq i < N} \frac{1}{i} = N_H - \frac{1}{N} < N_H$$

Quan es treballa amb aquestes sumatòries és cabdal no perdre en cap moment per quin domini es fa la sumatòria. En el darrer cas, al excloure N de la sumatòria es diu que $N_H - \frac{1}{N} < N_H$.

Si es suma la segona igualtat per $1 < i \leq N$ s'obté:

$$N_H - 1 = \sum_{1 < i \leq N} \frac{1}{i} < \sum_{1 < i \leq N} \int_{i-1}^{i} \frac{dx}{x} = \int_1^N \frac{dx}{x} = \ln N$$

Així doncs queden provats els dos marges de l'expressió i per tant retornant a l'anàlisi de l'algorisme es pot aplicar la demostració per $\sqrt{x}$.

□

El nombre de passos on es marquen compostos és:

$$N \sum_{p \leq \sqrt{N}} \frac{1}{p} = O(N \log N)$$

Tanmateix això no és del tot cert; en la demostració s'ha basat el límit superior en la consideració de N_H com una sèrie on entraven tots els nombres, primers o no, fins a $\sqrt{N}$.

Teorema 2.1.4 En una sumatòria sobre els primers es compleix

$$\sum_{p \leq N} \frac{1}{p} = \ln \ln N + o(1) + B_1$$

per una constant B_1.

Prova (Veure, teorema 427 [23]).

Per tant, com a algorisme per generar nombres primers grans és inviable ja que té una complexitat de $O(N \ln \ln N)$. Un test eficient hauria de prendre un temps polinòmic, aproximadament $O(\log N)$.

Obtinguda la complexitat de la funció es presenta un gràfic de la funció temps en relació a la potència de N. Resulta interessant veure que els resultats teòrics coincideixen força amb els pràctics.

$$f(x) = \frac{10^N \log_2 N \log_2 10}{10^8}$$

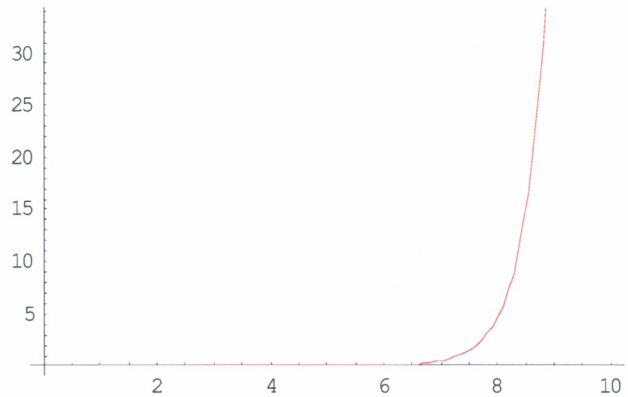


Figura 2.3

Abans de l'adveniment dels ordinadors aquells que es dedicaven a la teoria numèrica, a l'astronomia o a les matemàtiques en general i que havien de fer un treball numèric constant, disposaven de taules de factors que els ajudessin quan volien factoritzar un nombre.

Aquesta idea no ha quedat oblidada tot i que té un ús diferent; ja no es consulten taules de factors perquè poden ser infinites sinó que dins d'un algorisme que faci una funció determinada es demanen les factoritzacions de certs nombres. Executant el garbell d'Eratosthenes simplement canviant les posicions de la taula per enters enlloc de boleans no té un efecte significatiu en el temps d'execució simplement, es nota en la memòria perquè en lloc d'un bit per posició de la taula ara se'n ocupen 32 o 64 segons el rang d'enters amb el què el compilador permet treballar.

Quan es troba un factor s'emmagatzemem en la casella de manera que després ràpidament es pot resseguir la factorització. Per exemple la factorització de 1050: anant a la posició 1050 de la taula s'hi troba el 2, que es el primer factor, ara a la posició 1050/2 (525) hi ha el factor 3, a la posició 525/3 (175) hi ha el 5, es va a la posició (175/5) on s'hi troba un altra 5, al anar a la posició 7 s'hi troba un 0, és a dir, el 7 és primer.

$$1050 = 2 \cdot 3 \cdot 5^2 \cdot 7$$

Els temps d'execució de l'algorisme 2.1.2 són excessivament elevats per valors grans de N; la causa després d'una eternitat de proves es pot hipotetitzar però amb força elements de la nostra part, que es tracta d'un sobreeixement de la memòria *cache* de l'ordinador, aquella on emmagatzema informació constantment canviant o consultada. Hi ha programes que el factor limitant és la memòria, i aquest és un d'ells. (veure [1]).

Es proposa una altra versió del garbell d'Eratosthenes, aquest cop tenint en compte que tots els nombres primers són imparells i que per tant compleixen: $p \equiv \pm 1 \pmod 4$, d'aquesta manera es pot estalviar aproximadament la meitat de memòria i resultarà més eficient a mesura que N sigui major.

Teorema 2.1.5 El comportament assimptòtic de la següent sumatòria és:

$$\frac{1}{2} + 2 \sum_{p \le N, p \equiv 1 \pmod 4} p^{-1} = \ln \ln N + o(1)$$

Prova

Es remetem el lector a la bibiliografia [Davenport] perquè la demostració està fora del enfocament del treball.

Algorisme 2.1.6

```
BEGIN
 For i:=1 to sqrt(n) do
  Begin
   x:=2*i+3; // com a primer

   //eliminació dels múltiples imparells de
   //la forma

   x2:=(2*k+1)*x;     //per k en [i+1,n div i]
   si (x2< n*2) llavors t[(x2-3)div 2]:=false;
  End;
END.
```

En aquest cas s'ha optat per un pseudocodi molt allunyat de qualsevol llenguatge de programació perquè es vegi clarament la idea que s'ha desenvolupat, començar a introduir les variables no hagués estat més útil i el codi ja s'adjuntarà apart.

S'han utilitzat nombres representables com `2*i+3`, de manera que només poguessin ser senars i estalviéssim la meitat de la memòria. Cal ser força escèptics amb aquest mètode perquè tampoc permet cap millora considerable, en el sentit que la complexitat segueix sent la mateixa que en l'algorisme d'Eratosthenes, tant sols és aplicable en un rang d'entrada que es veurà tot seguit. En un principi el fet de dividir impedeix que pugui competir amb l'algorisme 2.1.1 però ràpidament el fet d'utilitzar menys memòria es girarà en el nostre favor.

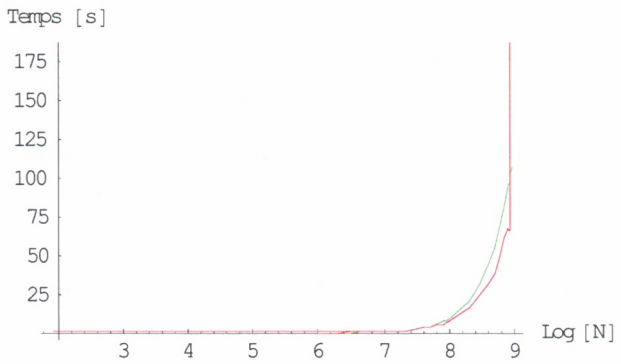


Figura 2.4

En aquest gràfic l'algorisme l'algorisme 2.1.1 apareix representat en vermell mentre que el 2.1.6 apareix en verd; es pot apreciar com el creixement del primer és inferior fins aproximadament $N = 10^9$. Però ara que es disposa d'una idea general cal veure de manera més precisa llur comportament, per aquesta raó s'ha prosseguit a experimentar amb diversos paràmetres d'entrada.

Una aproximació a l'interval $N = \left[10^6, 10^8\right]$ corroborarà el que la figura 2.4 suggeria.

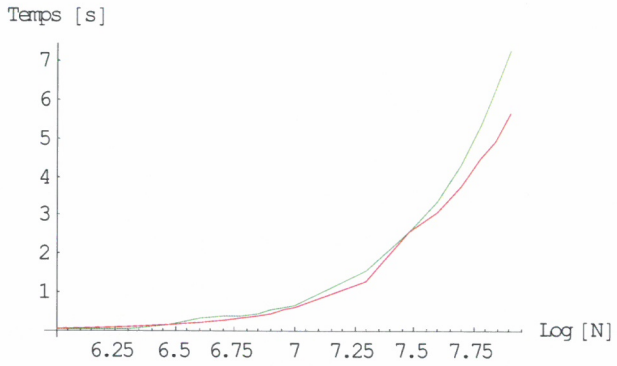


Figura 2.5

En l'interval $N = \left[10^8, 10^9\right]$ l'algorisme 2.1.3.6 va prenent territori pel seu menor ús de memòria i el fet que no consideri els nombres parells. S'ha determinat que definitivament el darrer programa basat en el garbell d'Eratosthenes resulta més eficient a partir de $N = 10^{8.95}$.

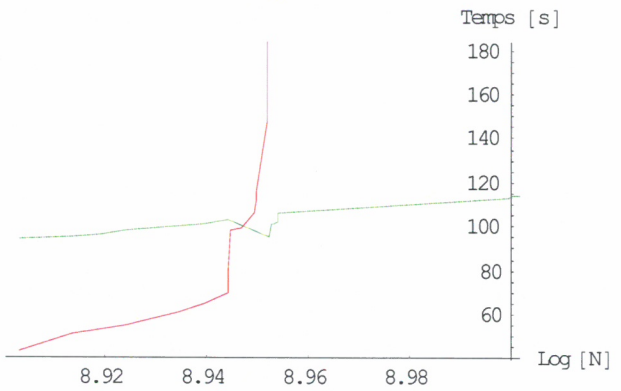


Figura 2.6

N	$\pi(N)$	**Temps**
896 500 000	45 839 615	0:1:35:20
999 999 000	50 847 489	0:1:53:68
$3 \cdot 10^9$	50 847 534	0:1:53:67
$1.05 \cdot 10^9$	53 257 350	0:1:59:85
$1.9 \cdot 10^9$	93 547 928	0:2:0:68
$2 \cdot 10^9$	98 222 287	0:2:1:54
$3 \cdot 10^9$	144 449 537	0:2:2:57

Taula 2.7

No s'ha pogut prosseguir a través d'aquest mètode perquè l'ordinador es queda sense memòria virtual el nombre de bits necessaris és $O(N)$.

2.2 Garbell d'Atkin

La idea és de A.O.L Atkin, professor ara *emeritus* de la universitat d'Illinois a Chicago, i D.J Bernstein, un jove programador i matemàtic. El seu article publicat a *Mathematics of computation* el 1999 porta com a títol, Prime sieves using irreducible quadratic forms. Abans d'aquest, Pritchard i Mairson ja s'havien adonat que es podia realitzar un garbell amb una complexitat de $O(N/\log\log N)$ però utilitzant $O(N^{1+o(1)})$ bits de memòria. Aquesta és una millora si es té en compte la complexitat dels garbells no combinatoris anteriors requerien $O(N)$ operacions i aproximadament $O(N^{1/2}(\log\log N/\log N))$ bits de memòria.

Atkin i Bernstein proposen un nou algorisme més eficient amb la mateixa complexitat $O(N/\log\log N)$ però que tant sols necessita $O(N^{1/2+o(1)})$ bits de memòria basat en la transformació d'un problema analític en un de geomètric. De fet pot ser interpretat com una generalització del garbell d'Eratosthenes tot i que aquest es basa estrictament en la definició de primer. Resulta convenient interpretar Eratosthenes però mantenint la idea essencial perquè després ens permetrà la generalització que es busca.

Definicions 2.2.1 $p^{\alpha}\|N$ denota la potència màxima d'un primer que divideix a N.

Per comptar divisors s'eliminen sempre els associats trivials. En $\mathbb{Z}$ els associats són $(-1)(-p)$ i $(-p)(-1)$. Si la norma d'una unitat és $|u|=1$ on $u\in\{\pm 1\}$ i $xy=N$ llavors:

$$(ux)\left(u^{-1}y\right)=N$$

En realitat els nombres primers també poden ser negatius però per convenció se'ls elimina.

Teorema 2.2.2 Un enter N lliure de quadrats es primer tant sols si i només si:

$$\#\{(x,y),\quad x,y>0 : xy=N\}\equiv 2(\mathrm{mod}\,4)$$

Prova

La demostració del primer teorema es dividirà en tres parts: la primera i la tercera són estrictament necessàries perquè posen en relació dos elements fonamentals. D'altra banda la segona completa la primera en el sentit que pot ser dificil de seguir el raonament si no s'especifiquen tots els passos.

S'estableix ara la primera relació a través de la funció Zeta per fer-li honor a Riemann tot i que Atkin i Bernstein hagin utilitzat un altra mètode.

$$\zeta^2(s) = \sum_{n\geq 1}\frac{1}{n^s}\sum_{m\geq 1}\frac{1}{m^s} = \sum_{n,m\geq 1}\frac{1}{(nm)^s}$$

$$= \sum_{k\geq 1}\frac{\sum_{mn=k} 1}{k^s} = \sum_{k\geq 1}\frac{\mathrm{d}(k)}{k^s}$$

En la primera línia la sumatòria a doble variable és el que a programació es diria *for* anidat. Primerament s'executa per m i quan aquesta arriba a infinit passa a n=2 i altra vegada m des de 1 a infinit. Cal veure que són sèries convergents per tota s superior a 1.

El resultat seria quelcom semblant a:

$$\left(1+\frac{1}{2^s}+\frac{1}{3^s}+\frac{1}{4^s}+\ldots\right)+\left(\frac{1}{2^s}+\frac{1}{4^s}+\frac{1}{6^s}+\ldots\right)+\left(\frac{1}{3^s}+\frac{1}{6^s}+\frac{1}{9^s}+\ldots\right)$$

El segon pas representa el nombre de vegades que apareix un mateix resultat, és a dir, el nombre de maneres que es pot escriure una mateixa k. Per exemple 2^{-s} només pot aparèixer de dues maneres diferents: al multiplicar 1 amb 2^{-s} i al revés. Tanmateix 6^{-s}, apareix de quatre maneres diferents com a producte de 1 amb 6^{-s}, 2^{-s} amb 3^{-s} i al revés. I com que el nombre de maneres que es pot formar k coincideix amb el nombre de divisors de k, s'obté la darrera igualtat.
Abans de prosseguir cal demostrar la següent igualtat ja que ens permetrà dur a terme el següent pas en el nostre raonament.

$$\prod_p\left(1-\frac{1}{p^s}\right)^{-1} = \prod_p\sum_{0\leq k}\frac{1}{p^{ks}}$$

Es pot seguir el següent raonament partint del producte d'Euler

$$\prod_p\left(1-\frac{1}{p^s}\right)^{-1} = \left(1-\frac{1}{p_1^s}\right)^{-1}\left(1-\frac{1}{p_2^s}\right)^{-1}\ldots$$

$$= \prod_p\sum_{0\leq k}\left(\frac{1}{p_i^s}\right)^k$$

$$= \left(1+\frac{1}{p_1^s}+\frac{1}{p_1^{2s}}+\frac{1}{p_1^{3s}}+\ldots\right)\left(1+\frac{1}{p_2^s}+\frac{1}{p_2^{2s}}+\frac{1}{p_2^{3s}}+\ldots\right)$$

$$= 1 + \sum_{1 \le i} \frac{1}{p_i^s} + \sum_{1 \le i \le j} \frac{1}{p_i^s p_j^s} + \sum_{1 \le i \le j \le k} \frac{1}{p_i^s p_j^s p_k^s} + \ldots$$

$$= 1 + \frac{1}{2^s} + \frac{1}{3^s} + \frac{1}{4^s} + \frac{1}{5^s} + \frac{1}{6^s} + \ldots = \zeta(s)$$

El pas més sobtat possiblement sigui el segon, per passar de la primera igualtat a la segona on s'han d'utilitzar sèries de Taylor. Es considera la funció $f(x) = (1-x)^{-1}$ i a través de sèries de Taylor s'observa que es pot expressar la funció com:

$$f(x) = f(x_0) + \frac{1}{1!} f'(x_0)(x - x_0) + \frac{1}{2!} f''(x_0)(x - x_0)^2 + \frac{1}{3!} f'''(x_0)(x - x_0)^3 + \ldots$$

A l'inici $x_0 = 0$ de manera que $f(x_0) = 1$. Amb la primera derivada s'obté:

$f'(x) = -(1-x)^{-2}$ i $f'(0) = -1$ però es tornarna tot positiu;

$f''(x) = 2(1-x)^{-3} = 2$; $f'''(x) = -6(1-x)^{-4} = -6 \ldots$ Al reconstruir s'obté:

$$f(x) = 1 + \frac{1}{1!} \cdot 1x + \frac{1}{2!} 2x^2 + \frac{1}{3!} 6x^3 + \ldots$$

$$= 1 + x + x^2 + x^3 + \ldots = \sum_{0 \le k} x^k$$

i substituint x per p^{-s} s'obté la identitat.

En el penúltim pas simplement comentar que la sumatòria infinita de sumatòries infinites és el mètode per denotar un producte de sumatòries infinites, en aquest cas el de les sèries geomètriques. A més des del punt de vista de la teoria dels nombres cada sumatòria és molt rellevant: en la primera apareixen tots els primers, en la segona aquells compostos de 2 factors primers, en la tercera els compostos de 3 factors primers i així fins a l'infinit.

Tant sols és una aplicació del Teorema Fonamental de l'Aritmètica. Ara finalment es repren la demostració del teorema 2.2.2 però tenint en compte la formula del producte d'Euler.

$$\zeta^2(s) = \prod_p \left(1 + p^{-s} + p^{-2s} + \ldots\right)^2 = \prod_p \left(\sum_{0 \le k} \left(p^{-s}\right)^k\right)^2$$
$$= \prod_p \left(1 - p^{-s}\right)^{-2}$$
$$= \prod_p \left(1 + \frac{2}{p^s} + \frac{3}{p^{2s}} + \ldots + \frac{n+1}{p^{ns}} + \ldots\right)$$
$$= \sum_{k \ge 1} \frac{\prod_{p^\alpha \| k} (1+\alpha)}{k^s}$$

Els dos primers passos no representen cap pas radical, m'he limitat a aplicar les igualtats que es coneixíen fins ara. Així doncs la tercera igualtat és la que reqereix més esforç per veure. Sigui $g(x) = (1-x)^{-2} = -f(x)$. Per tant si

$$f(x) = \sum_{n \ge 0} x^n$$
$$f'(x) = \sum_{n \ge 0} n x^{n-1} = \sum_{n \ge 0} (n+1) x^n$$

reemplaçant x per p^{-s} s'obté la tercera expressió. La darrera igualtat és una aplicació del Teorema Fonamental de l'Aritmètica i no admet comentaris.

En resum, si es té en compte aquest resultat a través de la formula del producte i l'anterior a través de la funció Zeta de Riemann es podrà constatar que:

$$d(k) = \prod_{p^\alpha \| k} (1+\alpha)$$

Aquest és un molt bon resultat ja que ens permet calcular els divisors de k. Si es té en compte la definició d'un nombre lliure de quadrats -tots els factors primers tenen multiplicitat 1- llavors és obvi que $d(k) = 2^{\delta(N)}$ on $\delta(N)$ és la quantitat de nombres primers diferents. Tot nombre primer complirà:

$$2^{\delta(N)} \equiv 2 \pmod 4$$

□

Exemple 2.2.3

$$\text{Divisors}\left(2^3 5^2 7^5 11^{23}\right) = 4 \cdot 3 \cdot 6 \cdot 24 = 1728$$

de la mateixa manera s'aplica la idea anterior al producte:

$$\frac{(j+1)(k+1)}{2^{js}3^{ks}} = \frac{(j+1)(k+1)}{\left(2^j 3^k\right)^s}$$

on el numerador representa la quantitat de maneres diferents que pot aparèixer un mateix nombre, és a dir, la quantitat de divisors que té. Al generalitzar per un enter $k = p_1^{\alpha_1} \ldots p_r^{\alpha_r}$ el nombre de divisors serà el producte de cada exponent més 1.

Definició 2.2.4 $\mathbb{Z}[\mathrm{i}]$ és un anell de Gauss que comprén tots els nombres de la forma $a + b\mathrm{i}$ on $a, b \in \mathbb{Z}$. Les unitats del anell seran quatre: $u \in \{\pm 1, \pm \mathrm{i}\}$, permetran a partir d'un sol nombre complex formar-ne 7 més associats. Bé això no és del tot cert, a partir d'un nombre complex δ es generen 3 més i a partir del seu conjugat, $\overline{\delta}$, les altres quatre. Pels vuit nombres la seva norma, el mòdul al quadrat, serà la mateixa. Per cada unitat u, es diu que $u\delta$ és un associat de δ.

Lema 2.2.5 Els primers de $\mathbb{Z}[\mathrm{i}]$ són $(1+\mathrm{i})$, $q \equiv 3 \pmod 4$ on $q \in \mathbb{Z}$ i és primer. També tot nombre π tal que $N(\pi) = p \equiv 1 \pmod 4$.

Prova

S'han diferenciat tres casos perquè a partir d'aquests es poden englobar tots els primers de l'anell $\mathbb{Z}[\mathrm{i}]$. El primer cas és una excepció, un cas estrany fins i tot en complexes. Ja en enters és l'únic primer parell, doncs en el $\mathbb{Z}[\mathrm{i}]$ és l'únic associat al seu conjugat, per tant només es pot generar tres nombres complexos més a través de les unitats. Per veure el segon cas es suposa que $q \equiv 3 \pmod 4$ no és primer i s'arriba a una contradicció. Sigui :

$$q = (a + b\mathrm{i})(c + d\mathrm{i})$$
$$q^2 = \left(a^2 + b^2\right)\left(c^2 + d^2\right)$$

tenint en que no hi ha cap nombre congruent amb tres mòdul quatre expressable com a suma de quadrats perquè $a^2 + b^2 \equiv \{0,1,2\} \pmod 4$; llavors només hi ha dos casos possibles en què es pot donar la factorització anterior:

$$a^2 + b^2 = 1 \rightarrow c^2 + d^2 = q^2$$
$$a^2 + b^2 = q^2 \rightarrow c^2 + d^2 = 1$$

Finalment si $a + b\mathrm{i}$ és primer, llavors $N(a + b\mathrm{i})$ no pot ser compost perquè si $p \mid N(a + b\mathrm{i}) \rightarrow \pi \mid a + b\mathrm{i}$ llavors perquè :

$$(a + b\mathrm{i})(a - b\mathrm{i}) = \pi\overline{\pi}k = pk = N(a + bi)$$

per tant $\pi \mid a + b\mathrm{i}$.

□

Teorema 2.2.6 Un enter $N \equiv 1 \pmod 4$ lliure de quadrats és primer si

$$\#\left\{(x > 0, y > 0) \mid x^2 + y^2 = N\right\} \equiv 2 \pmod 4$$

Es pot representar d'aquesta manera els nombres amb la forma:
$60k + d$, $d \in \{1,13,17,29,37,41,59,53\}$. El nombre de solucions ha de ser dos congruent amb quatre perquè no s'han eliminat les simetries en aquesta forma, per contra Atkin i Bernstein desenvolupen el seu algorisme a través de $4x^2 + y^2$.

Prova: utilitzen la forma quadràtica $x^2 + y^2 = N$, factoritzable en $\mathbb{Z}[\mathrm{i}]$ de la següent manera:

$$x^2 + y^2 = (x + y\mathrm{i})(x - y\mathrm{i})$$

Es pot aplicar el Teorema Fonamental de l'Aritmètica en aquest anell també de manera que:

$$\delta = \pi_1^{\alpha_1} \cdots \pi_r^{\alpha_r}$$

on π són els primers d'aquest anell tal que si $uv = \pi$ llavors la norma de u o la norma de v sigui igual a ± 1. Per la demostració cal tenir en ment el lema 2.1.3.5 perquè defineix els primers de l'anell $\mathbb{Z}[\mathrm{i}]$.

Sigui

$$\begin{aligned}\theta_n &= \#\{\delta : N(\delta) = n\} \\ &= \#\{x^2 + y^2 = n : (x > 0, y > 0)\}\end{aligned}$$

és a dir la quantitat de nombres complexes que tenen la norma n, eliminant les simetries de les unitats. Es fa el mateix plantejament de la funció Zeta que ha permès demostrar el primer lema d'aquest capítol.

$$\zeta_{\mathbb{Z}[\mathrm{i}]}(s) = \sum_{n\geq 1}\frac{\theta_n}{n^s} = \prod_{\pi}\left(1-\frac{1}{N(\pi)^s}\right)^{-1} \tag{1}$$

$$= \left(1-\frac{1}{2^s}\right)^{-1}\prod_{p\equiv 1(\bmod 4)}\left(1-\frac{1}{p^s}\right)^{-2}\prod_{q\equiv 3(\bmod 4)}\left(1-\frac{1}{q^{2s}}\right)^{-1} \tag{2}$$

$$= \zeta(s)\prod_{p\equiv 1(\bmod 4)}\left(1-\frac{1}{p^s}\right)^{-1}\prod_{q\equiv 3(\bmod 4)}\left(1+\frac{1}{q^s}\right)^{-1} \tag{3}$$

$$= \zeta(s)\prod_{p}\left(1-\frac{\chi(p)}{p^s}\right)^{-1} = \zeta(s)\sum_{n\geq 1}\frac{\chi(n)}{n^s} \tag{4}$$

$$= \sum_{m\geq 1}\frac{1}{m^s}\sum_{n\geq 1}\frac{\chi(n)}{n^s} = \sum_{k=mn\geq 1}\frac{\sum_{d|k}\chi(d)}{k^s} \tag{5}$$

$$\therefore \quad \theta_{\mathrm{k}} = \sum_{d|k}\chi(d) = \prod_{p^{\alpha}\| k}\left(1+\chi(p)+\cdots+\chi\left(p^{\alpha}\right)\right) \tag{6}$$

Si $\alpha = 1$ i $q|k$ llavors $\theta_k = 0$ si no $\theta_k = 2^{\delta(n)} \equiv 2(\bmod 4)$, si i només si n és primer. $\chi(n)$ és una funció multiplicativa, anomenada caràcter Dirichlet, que es demostra iterant, l'única condició és que els factors han de ser coprimers perquè la funció no és completament multiplicativa. És a dir:

$$\chi\left(p_1^{\alpha_1}\cdots p_r^{\alpha_r}\right) = \chi\left(p_1^{\alpha_1}\right)\chi\left(p_2^{\alpha_2}\cdots p_r^{\alpha_r}\right)$$

En la demostració χ val -1 si $q \equiv 3(\bmod 4)$, 1 si $p \equiv 1(\bmod 4)$ i 0 si n=2.

El nou desenvolupament de la funció Zeta no implica cap canvi conceptual radical, és simplement l'adaptació d'aquesta a les característiques dels primers en l'anell $\mathbb{Z}[i]$. Per veure la primera igualtat (1) cal recordar la veritable definició de la funció zeta com a sèrie Dirichlet; en $\mathbb{Z}$ el numerador és sempre 1 perquè, eliminant simetries, només existeix un nombre tal que la seva norma sigui n.

(1) és l'aplicació de la formula del producte d'Euler però en $\mathbb{Z}[i]$. No s'inclou la demostració de que també es compleix en $\mathbb{Z}[i]$ perquè és quasi idèntica que per enters. Ara bé, què canvia? Doncs els p es converteixen en π, i com que no es necessita la variable complexa i perquè es considera θ_n, s'aplica la norma a π.

(2) de primers n'hi ha tres tipus, el 2, els $N(\delta) = p \equiv 1 \pmod 4$ i els $q \equiv 3 \pmod 4$. Els congruents amb 1 mòdul quatre tenen dos primers de $\mathbb{Z}_{[i]}$ per cada norma: π i $\overline{\pi}$, per això el producte està elevat al quadrat. Els de la forma q+0i, tenen norma q^2, per això cada primer quedarà elevat al quadrat.

(3) és una reordenació dels termes. Només hi ha dos tipus de primers mòdul quatre, els que tenen residu 1 i els de residu 3. D'aquí segueix immediatament el fet que:

$$(2) = \left(1-\frac{1}{2^s}\right)^{-1} \prod_{p\equiv 1 \pmod 4}\left(1-\frac{1}{p^s}\right)^{-1} \prod_{p\equiv 1 \pmod 4}\left(1-\frac{1}{p^s}\right)^{-1} \prod_{q\equiv 3 \pmod 4}\left(1-\frac{1}{q^s}\right)^{-1} \prod_{q\equiv 3 \pmod 4}\left(1+\frac{1}{q^s}\right)^{-1}$$
$$= \zeta(s) \prod_{p\equiv 1 \pmod 4}\left(1-\frac{1}{p^s}\right)^{-1} \prod_{q\equiv 3 \pmod 4}\left(1+\frac{1}{q^s}\right)^{-1} = (3)$$

(4) representa la introducció del caràcter Dirichlet que s'ha comentat abans: és per sintetitzar l'expressió.

(5) El producte de les dues sèries es fa terme a terme.

(6) segueix al fet que si $\chi(k)$ és multiplicativa llavors $\sum_{d|k} \chi(d)$ també ho és.

Es veu ara com s'ha desenvolupat l'expressió:

$$\sum_{d|k}\chi(d)=\prod_{p^\alpha\|k}\sum_{d|p^\alpha}\chi(d)=\prod_{p^\alpha\|k}\left(1+\chi(p)+\cdots+\chi\left(p^\alpha\right)\right)$$

□

De manera anàloga es poden demostrar els teoremes següents.

Teorema 2.2.7
Un enter $N\equiv 1(\mathrm{mod}\,6)$ lliure de quadrats és primer si

$$\#\left\{(x>0,y>0)\ \middle|\ 3x^2+y^2=N\right\}\equiv 1(\mathrm{mod}\,2)$$

Amb aquesta forma quadràtica, es representen tots els nombres de la forma $60k+d$, on $d\in\{1,7,13,19,31,37,43,49\}$.

Teorema 2.2.8
Un enter $N\equiv 11(\mathrm{mod}\,12)$ lliure de quadrats és primer si

$$\#\left\{(x>0,y>0)\ \middle|\ 3x^2-y^2=N\right\}\equiv 1(\mathrm{mod}\,2)$$

representen tots els nombres de la forma $60k+d$, on $d\in\{11,23,47,59\}$

Per tant Atkin i Bernstein dissenyen una criba similar a la d'Eratosthenes amb la diferència que en lloc de considerar els nombres de la forma xy utilitza tres formes quadràtiques irreductibles que engloben a tots els nombres primers. A banda de demostrar aquests teoremes fan una interpretació geomètrica del problema, tractant els diferents valors de x i y com punts en una cònica d'un retícul de coordenades enteres.

El problema inicial de comptar primers queda dividit en tres parts similars però que consideren nombres de diferent naturalesa i a més es transforma en un problema geomètric. Aquí en la memòria del treball només inclou l'algorisme que detecta els primers representables com x^2+y^2 perquè els altres són molt similars. Evidentment quan es presenti la comprativa entre Atkin i Eratosthenes s'hauran implementat les tres parts.

Algorisme 2.2.9
OUTPUT: retorna tots els primers de la forma
$60k+d$ quan d pertany {1,13,17,29,37,41,49,53} en l'interval $L \leq k < L+B$

BEGIN
T[L]=T[L+1]=···=T[L+B]=0;
1. Per cada (x,y,k) tal que $x>0$, $y>0$, $L \leq k < L+B$ i $4x^2+y^2=60k+d$
Do: T[k]:=1-T[k];

2. Per cada $q \geq 7$ i $q^2<60L+B$;
Per cada k amb $60k+d$ divisible per q^2
T[k]:=0

3. Retorna $60k+d$, per cada k tal que T[k]=1

END.

El primer pas conta el nombre de parelles x i y que permetin representar a $60k+d$; el valor de l'array va canviant de zero a 1 per cada solució que es troba perquè es recorda que pel teorema 1, el nombre de solucions ha de ser senar. La segona part elimina tots els nombres que no siguin lliures de quadrats: d'aquesta manera només resten els primers.

L'anàlisi de la complexitat d'aquest algorisme es divideix en dues parts en realitat; el pas 1 i el pas 2. De fet el primer pas és un altre algorisme geomètric que ara es comentarà tot seguit. La segona part sí admet directament unes cotes per aproximar el nombre d'iteracions.

Cal veure quants múltiples de q^2 hi ha fins a B: $q^2 k \leq B$, és a dir: $k \leq \frac{B}{q^2}$. Per trobar el nombre de iteracions cal sumar per totes les $q \leq \sqrt{60L+B}$ queda doncs:

$$\sum_{q \leq \sqrt{60L+B}} \frac{B}{q^2} = B \sum_{q \leq \sqrt{60L+B}} \frac{1}{q^2} \leq B \sum_{n \leq \sqrt{60L+B}} \frac{1}{n^2} \leq B \sum_{n \geq 1} \frac{1}{n^2} = B\zeta(2) = B\frac{\pi^2}{6}$$

q denota nombres primers i n qualsevol enter de manera que la justificació de les cotes és immediata. La última igualtat no es demostrarà aquí, és a títol de curiositat i va rebre el nom de problema de Basel, ciutat suïssa on els dos germans Bernoulli, Jakob (1687-1705) i Johan (1705-1748), van impartir classes de matemàtiques. En realitat s'hauria de dir problema de Bologna perquè va ser proposat per primer cop per Pietro Mengoli el 1644 tanmateix van ser els germans Bernoulli qui el van fer conegut el 1689. En el llibre que Jakob va publicar incloïa

demostracions del seu germà i seves respecte la divergència de les sèries harmòniques però demanava a qualsevol que pogués donar una expressió tancada pel quadrat d'aquestes que els ho fes saber. El jove Leonhard Euler va donar la solució el 1735, 46 anys més tard.

L'algorisme 2.2.9b és imprescindible pel funcionament del primer; compta el nombre de parelles de x i y tal que $4x^2 + y^2 = 60k + d$. En realitat aquesta forma quadràtica és una hipèrbole però tant sols ens interessen els valors d'aquesta amb coordenades enteres positives; si s'imagina la figura, només se'n necessita una quarta part d'aquesta. Aquest és el mètode de comptar punts utilitzat per Atkin i Bernstein: partint del primer quadrant en un rang definit entre $60L \leq 4x^2 + y^2 < 60L + B$ compta punts de la retícula; s'ha d'imaginar dues el·lipses que marquen el límit i l'àrea del mig representant tots els punts, tant els de coordenades enteres com els reals. Aquest és l'algorisme que serveix per enumerar els punts en la retícula de coordenades enteres.

Algorisme 2.2.9.b

INPUT: donats $d<60$, $f \leq 15$, $g \leq 30$ tal que $4f^2+g^2 \equiv d \pmod{60}$

OUTPUT: retorna les tripletes (x,y,k) amb $x>0, y>0$, $L \leq k < L+B$, $4x^2+y^2=60k+d$, $x \equiv f \pmod{15}$, $y \equiv g \pmod{30}$

BEGIN

1 $x:=f$; $y_0:=g$; $k_0:=(4f^2+g^2-d)/60$.
(Si es comença en el pas (3) ens estaríem desplaçant de dreta a esquerra quan $k_0:=(4f^2+y_0^2-d)/60$)

2 si $k_0<L+B$ llavors $k_0:=k_0+2x+15$; $x:=x+15$; ves a (2);

3 $x:=x-15$; $k_0:=k_0-2x-15$; para si $x \leq 0$ (ves a l'esquerra)

4 si $k_0<L$: $k_0:=k_0-2x-15$. Para si $x \leq 0$ (Puja si cal)
ves a (4)

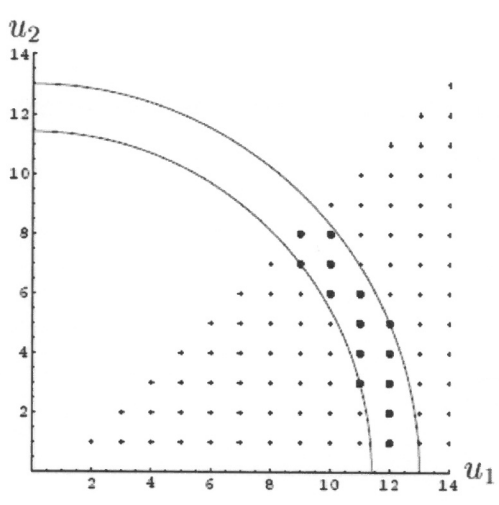
5 ($4x^2+y_0^2 \geq 60L$ si $y_0>30$ llavors $4x^2+(y_0+30)^2<60L$)
$k:=k_0$; $y:=y_0$;

6 ($4x^2+y_0^2=60k+d \geq 60L$, si $k< L+B$)
retorna (x,y,z). $k:=k+y+15$; $y:=y+30$; ves a (6)

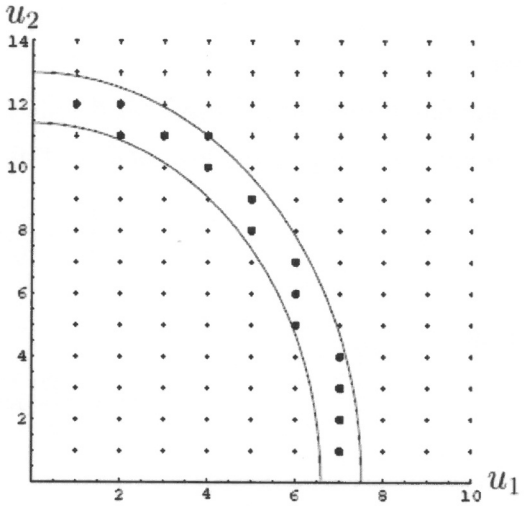
7 ves a (3)

END.

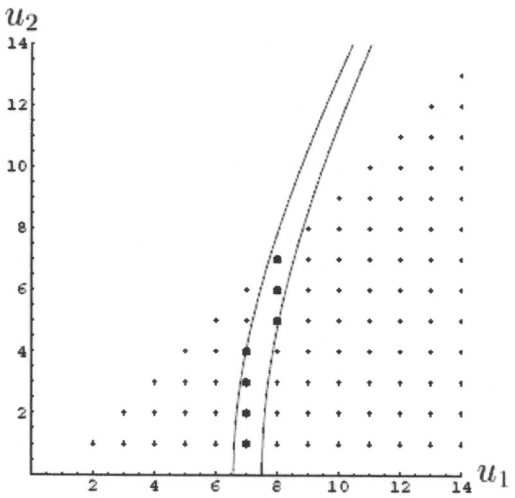
Abans de prosseguir s'haurà de provar l'àrea de l'el·lipse.



a. $N \equiv 1 (\mathrm{mod}\, 4)$, $N = u_1^2 + u_2^2$



b. $N \equiv 7 (\mathrm{mod}\, 12)$, $N = 3u_1^2 + u_2^2$



c. $N \equiv 11 (\mathrm{mod}\, 12)$, $N = 3u_1^2 - u_2^2$

Figura 2.8 Mostra una representació de les tres formes irreductibles quadràtiques. Les solucions d'aquestes són el conjunt de coordenades enteres en un retícul limitat per dues el·lipses.

Teorema 2.2.10 : L'àrea de l'el·lipse és πab

Prova

Per calcular l'àrea de l'el·lipse cal d'aplicar càlcul, primer expressant y en termes de x tenint en compte que la funció de l'el·lipse és:

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$$

per tant:

$$y=\pm\frac{b}{a}\sqrt{a^2-x^2}\ .$$

Com que es considera que l'elipse està centrada, es pot aplicar el principi de simetria: calcular l'àrea d'aquesta en el primer quadrant i multiplicar per quatre.

$$A=4\frac{b}{a}\int_0^a\sqrt{a-x^2}dx$$

no és una funció fàcil d'integrar, s'haurà d'aplicar el mètode de substitució. Ara $x=a\sin\theta$ i $dx=a\cos\theta$; adaptant els límits d'integració s'obtè:

$$\sqrt{a^2-x^2}=\sqrt{a^2-a^2\sin^2\theta}=\sqrt{a^2\cos^2\theta}=a\left|\cos\theta\right|$$

$$A=4\frac{b}{a}\int_0^a\sqrt{a-x^2}dx=A=4\frac{b}{a}\int_0^{\pi/2}\left(a\cos\theta\right)^2d\theta$$

$$=4ab\int_0^{\pi/2}\cos^2\theta d\theta=4ab\int_0^{\pi/2}\tfrac{1}{2}+\tfrac{1}{2}\cos 2\theta d\theta$$

$$=2ab\left[\theta+\tfrac{1}{2}\sin 2\theta\right]_0^{\pi/2}=2ab\frac{\pi}{2}=\pi ab$$

No s'han tingut en compte en cap moment el valor absolut perquè sempre es treballa amb valors positius dins l'interval $0\le\theta\le\pi/2$.

□

Retornant al cas que interessa; l'el·lipse de forma $\frac{x^2}{15B}+\frac{y^2}{60B}=1$, com que $a=\sqrt{15B}$ i $b=\sqrt{60B}$,

llavors

$$\pi\sqrt{15B}\sqrt{60B} = \pi\sqrt{5\cdot 3\cdot B}\sqrt{5\cdot 3\cdot 2^2\cdot B} = 30B\pi$$

però com que només es considera el primer quadrant: el nombre de punts a tenir en compte -enters i reals- (àrea) és

$$\frac{30\pi B}{4} = \frac{15B}{2} = \frac{\pi}{8}60B\,.$$

Encara es pot limitar més els punts a considerar i per tant fer una millor aproximació al comportament asimptòtic de l'algorisme. Com que x i y estan elevats al quadrat i x està multiplicat per quatre, valen x mòdul 15 i y mòdul 30. Es compta doncs en una congruència de 15 per x i una de 30 per y.

$$\begin{aligned}
&x \equiv x'\,(\mathrm{mod}\,15)\\
&4(x+15)^2 = 4x^2+120x+30^2 \equiv 4x^2\,(\mathrm{mod}\,60)\\
&y \equiv y'\,(\mathrm{mod}\,30)\\
&(y+30)^2 = y^2+60y+30^2 \equiv y^2\,(\mathrm{mod}\,60)
\end{aligned}$$

De manera que es pot dir que el nombre de punts a considerar és

$$\frac{1}{450}\frac{\pi}{8}60B = \frac{60\pi B}{3600} = \frac{\pi B}{60}.$$

Per estudiar el comportament asimptòtic de la criba d'Atkin i Bernstein cal fer unes observacions prèvies, sigui:

$$\begin{aligned}
W &= 12\prod_{p<\sqrt{\ln N}} p\\
\ln W &= \sum_{p<\sqrt{\ln N}} \ln p + \ln 12 \simeq \pi\left(\sqrt{\ln N}\right) < \sqrt{\ln N}
\end{aligned}$$

Els primers es conten en intervals $[WL, WL+WB]$ utilitzant $\varphi(W)B$ operacions, on $\varphi(W)$ denota la funció totient d'Euler i conta la quantitat de nombres coprimers amb W.

Sigui B un enter al voltant de $W\sqrt{N}$,

$$\varphi(W) = W\prod_{p|W}\left(1-\tfrac{1}{p}\right) = W\prod_{p<\sqrt{\ln N}}\left(1-\tfrac{1}{p}\right) = O\left(\frac{W}{\ln\ln N}\right)$$

perquè

$$\sum_{p<N}\ln\left(1-\tfrac{1}{p}\right) \sim \sum_{p<N}\frac{1}{p} \sim \int_2^N \frac{dt}{t\ln t} = \int_{\ln 2}^{\ln N}\frac{1}{u}du \simeq \ln\ln N$$

(en la darrera igualtat s'ha aplicat la substitució; $u = \ln t$ i $t^{-1}dt = du$).

En total l'algorisme té una complexitat de

$$\frac{N}{WB}\varphi(W)B \sim \frac{N}{\ln\ln N}$$

on WB denota la quantitat d'elements en un interval, N/WB les iteracions per interval, mentre que $\varphi(W)B$ és la quantitat d'operacions que es necessiten per interval.

Finalment s'ha optat per un mètode d'enumeració dels punts del retícul menys refinat però força eficient que troba les solucions de cada forma quadràtica en l'interval $[x_1...x_2]$, tal que $3 < x_1 < x_2$, de la següent manera:

```
For (u1=Ceiling(Sqrt(x1/2)); u1^2≤x2; u1++)
  For (u2=Ceiling(Sqrt(max(0,x1-u1^2))); (u1<u2)&(n= u1^2-u1^2)≤x2; u2++)
    If (n mod 4=1)then t[n]:=t[n]-1;

For (u1=1; 3u1^2≤x2; u1++)
  For (u2=Ceiling(Sqrt(max(0,x1-3u1^2)));(n=3u1^2-u1^2)≤x2; u2++)
    If (n mod 12=7)then t[n]:=t[n]-1;

For (u1=Ceiling(Sqrt(x1/3)); 2u1^2≤x2; u1++)
  For (u2=Ceiling(Sqrt(max(0,3u1^2-x2)); (u2<u1)&(n= 3u1^2-u1^2)≥x2; u2++)
    If (n mod 4=1)then t[n]:=t[n]-1;
```

Resultats experimentals 2.2.11

Per poder veure l'eficiència de l'algorisme es recomana descarregar i executar l'aplicació programada per Bernstein mateix i disponible a http://cr.yp.to/primegen/primegen-0.97.tar.gz.

2.3 Garbell de Legendre

El 1808 Antoine Marie Legendre (1752-1833) va proposar una formulació combinatòria del garbell d'Eratosthenes que permet estalviar molta memòria. El mètode de Legendre permet calcular $\pi(N)$ si es coneixen els primers menors o iguals que $\sqrt{N}$.

Sigui A el conjunt d'enters en l'intèrval $[1..X]$:

$$A = \{N \in \mathbb{Z} : 1 \leq N \leq X\}$$

i sigui

$$P = \prod_{p<z} p$$

on $z = \lfloor \sqrt{N} \rfloor$. El garbell es defineix com:

$$S(A,P) = \{N \in \mathbb{Z} : 1 \leq N \leq X, \text{mcd}(N,P) = 1\}$$

és a dir els nombres entre 1 i X que no tenen factors més petits que z.

Teorema 2.31 de Legendre

$$\pi(X) - \pi(\sqrt{X}) + 1 = \sum_{d|P(z)} \mu(d) \left\lfloor \frac{X}{d} \right\rfloor = S(A,P)$$

on μ denota la funció de Möbius i $\lfloor\ \rfloor$ la part entera.

Definició 2.3.2 La funció de Möbius es calcula de la següent manera:

$$\mu(1) = 1$$
$$\mu(d) = 0 \text{ si } p^2 | d \text{ per un primer } p$$
$$\mu(p_1 p_2 \cdots p_r) = (-1)^r \text{ si } p_1, \ldots, p_r \text{ són primers diferents}$$

Corol·lari 2.3.3 El garbell es pot rescriure com:

$$S(A,P) = \lfloor X \rfloor - \sum_{p_1<z} \left\lfloor \frac{X}{p_1} \right\rfloor + \sum_{p_1<p_2<z} \left\lfloor \frac{X}{p_1 p_2} \right\rfloor - \sum_{p_1<p_2<p_3<z} \left\lfloor \frac{X}{p_1 p_2 p_3} \right\rfloor + \cdots$$
$$= \sum_{d|P(z)} \mu(d) \left\lfloor \frac{X}{d} \right\rfloor$$

Tot seguit es prossegueix a reinterpretar el garbell d'Eratosthenes tal i com Legendre ho va fer per poder comprendre millor l'expressió que defineix el seu garbell. Donat un conjunt $S = \{1,..,X\}$, C_i denota els subconjunts de S que contenen els múltiples dels primers menors que $\sqrt{X}$. Els primers de S seran els primers menors que $\sqrt{X}$ i tots els elements de S que no pertanyin a la unió de $C_1 \cup C_2 \cup ... \cup C_r$. Fins aquí no hi ha res de nou, és una explicació del garbell d'Eratosthenes des del punt de la teoria de grups.

Definició 2.3.4 Generalitzant pel conjunt S, on $p_1, p_2, ..., p_r \leq \sqrt{N}$ establim que $C_i = \{a \in S : p_i | a\}$.

Definició 2.3.5 Pels valors $j_1, j_2, ..., j_k$ sigui $N(j_1, j_2, ..., j_k) = \#(C_{j_1} \cap C_{j_2} \cap ... \cap C_{j_k})$, és a dir, el nombre d'elements de S divisibles per $p_{j_1} \cdots p_{j_k}$ perquè els p_j són primers diferents. De manera que:

$$S_k = \sum N(j_1, j_2, ..., j_k)$$

i $S_0 = \#S = \lfloor X \rfloor - 1$.

El fet de reformular un problema sovint té avantatges perquè es pot aprofitar eines diferents per resoldre el problema inicial, en aquest cas el principi d' inclusió i exclusió.

Teorema 2.3.6 El nombre d'elements que pertanyen a S i no a cap C_i és

$$S_0 - S_1 + S_2 - \cdots + (-1)^k S_k = \pi(X) - \pi\left(\sqrt{X}\right)$$

Prova

Sigui a un enter que pertany exactament a $N \geq 1$ dels conjunts C_i i que per tant té n factors primers diferents. S'ha de veure quants cops es compta en els successius garbells parcials, és a dir en S_i. En S_0 un cop perquè és un cas especial; en S_1 només n cops, un per cada un dels subconjunts als què pertany.

En S_2 s'utilitzen parelles de subconjunts, i es conta un cop per cada vegada en què apareix en dos subconjunts dels N totals, per tant $\binom{N}{2}$. Generalitzant l'expressió per S_i s'obté que l'enter a es conta

$$1-\binom{N}{1}+\binom{N}{2}-\ldots+(-1)^N\binom{N}{N}$$

cops. Si w no pertany a cap dels C_i llavors només es conta en S_0. Els elements de S que no pertanyen als C_i, no són divisibles per cap primer menor o igual que $\lfloor\sqrt{X}\rfloor$, són per tant aquells nombres primers p tal que $\lfloor\sqrt{X}\rfloor < p \le X$. Per tant

$$S_0 - S_1 + S_2 - \cdots + (-1)^k S_k = \pi(X) - \pi\left(\sqrt{X}\right)$$

□

Teorema 2.3.7 El nombre de termes majors que zero de les sumatòries 2.3.3 és exactament

$$\frac{6(1-\ln 2)}{\pi^2}X \approx O(X)$$

Prova

La demostració està fora l'enfocament del treball i es remet el lector a [32].

Exemple 2.3.8 Sigui $X = 43$ i $\lfloor\sqrt{X}\rfloor = 6$. Els primers p tal que $p \le 6$ són 2, 3 i el 5. S'estableix la següent notació:

$N(1) =$ Nombre d'elements en $\{1,\ldots,X\}$ divisibles entre 2.

$N(1,2) =$ Nombre d'elements en $\{1,\ldots,X\}$ divisibles entre 2 i 3.

S'inicia ara el garbell 2.3.3

$$S_0 = \{1,\ldots,X\} = X - 1 = 42$$

$$S_1 = N(1) + N(2) + N(3) = \left\lfloor\tfrac{43}{2}\right\rfloor + \left\lfloor\tfrac{43}{3}\right\rfloor + \left\lfloor\tfrac{43}{5}\right\rfloor = 21 + 14 + 8 = 43$$

$$S_2 = N(1,2) + N(1,3) + N(2,3) = \left\lfloor\tfrac{43}{6}\right\rfloor + \left\lfloor\tfrac{43}{10}\right\rfloor + \left\lfloor\tfrac{43}{15}\right\rfloor = 7 + 2 + 4 + 13$$

$$S_3 = N(1,2,3) = \left\lfloor\tfrac{43}{30}\right\rfloor = 1$$

$$\pi(43) - \pi\left(\sqrt{43}\right) = S_0 - S_1 + S_2 - S_3 = 42 - 43 + 13 - 1 = 11$$

$\pi(43) - 3 = 11 \rightarrow$ de manera que $\pi(43) = 14$

Exemple 2.3.9 S'ha considerat important proveir al lector amb exemples per assegurar que es comprenia bé el significat de l'expressió assolida per Legendre. Tot seguit es veurà com es van generant les diferents permutacions amb els primers.

$$\begin{aligned}\pi(100) &= \pi(10) + 99 - \left\lfloor \tfrac{100}{2} \right\rfloor - \left\lfloor \tfrac{100}{3} \right\rfloor - \left\lfloor \tfrac{100}{5} \right\rfloor - \left\lfloor \tfrac{100}{7} \right\rfloor + \\ &= \left\lfloor \tfrac{100}{2\cdot 5} \right\rfloor + \left\lfloor \tfrac{100}{2\cdot 7} \right\rfloor + \left\lfloor \tfrac{100}{3\cdot 5} \right\rfloor + \left\lfloor \tfrac{100}{3\cdot 7} \right\rfloor + \left\lfloor \tfrac{100}{5\cdot 7} \right\rfloor - \left\lfloor \tfrac{100}{2\cdot 3\cdot 5} \right\rfloor - \\ &= \left\lfloor \tfrac{100}{2\cdot 3\cdot 7} \right\rfloor - \left\lfloor \tfrac{100}{2\cdot 5\cdot 7} \right\rfloor - \left\lfloor \tfrac{100}{3\cdot 5\cdot 7} \right\rfloor + \left\lfloor \tfrac{100}{2\cdot 3\cdot 5\cdot 7} \right\rfloor = \\ &= 4 + 99 - 50 - 33 - 20 - 14 + 16 + \\ &10 + 7 + 6 + 4 + 2 - 3 - 2 - 1 - 0 + 0 = 25\end{aligned}$$

Exemple 2.3.10 Ara es farà una especial atenció a com es tracten els primers en un garbell per un valor de X prou gran. Sigui $X = 729$ i $\sqrt{X} = 27$. Per calcular S_4 s'han de considerar 4 primers menors que 27.

$$2 \cdot 3 \cdot 5 \cdot 7 = 210$$
$$2 \cdot 3 \cdot 5 \cdot 11 = 330$$
$$2 \cdot 3 \cdot 5 \cdot 13 = 390$$
$$2 \cdot 3 \cdot 5 \cdot 17 = 510$$
$$2 \cdot 3 \cdot 5 \cdot 19 = 570$$
$$2 \cdot 3 \cdot 5 \cdot 23 = 690$$

mentre el producte de quatre primers és menor a X s'augmenta el darrer primer. Ara s'augmenta el tercer primer.

$$2 \cdot 3 \cdot 7 \cdot 11 = 462$$
$$2 \cdot 3 \cdot 7 \cdot 13 = 546$$
$$2 \cdot 3 \cdot 7 \cdot 17 = 714$$

El següent primer ja és massa gran de manera que no s'ha considerat. En lloc d'utilitzar el 7 ara es considera l'onze.

$$2 \cdot 3 \cdot 11 \cdot 13 = 858$$

Aquest torna a ser massa gran i no cal seguir provant. S'ha d'augmentar el segon primer:

$$2 \cdot 5 \cdot 7 \cdot 11 = 770$$

També és massa gran, s'augmenta el primer nombre primer de la llista.

$$3 \cdot 5 \cdot 7 \cdot 11 = 1155$$

Supera el valor de X de manera que tampoc es considera; s'ha acabat d'enumerar les seqüencies de primers a considerar en S_4.

Aquest mètode és cabdal perquè tots els mètodes utilitzats actualment es basen en optimitzacions d'aquest que és el primer mètode combinatori per calcular $\pi(x)$. El codi de Pascal resultat de la interpretació de la formula 2.3.3 de Legendre ha permès calcular valors de $\pi(N)$ per valors de N força elevats.

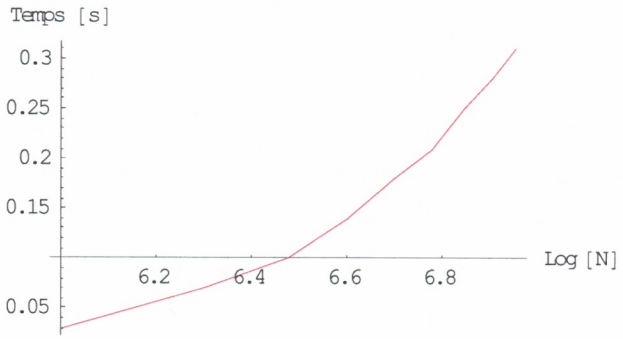


Figura 2.9

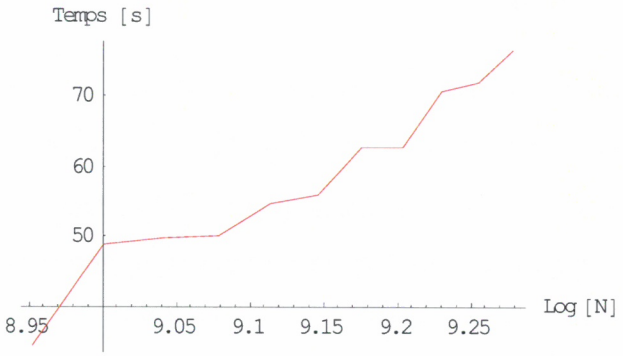


Figura 2.10

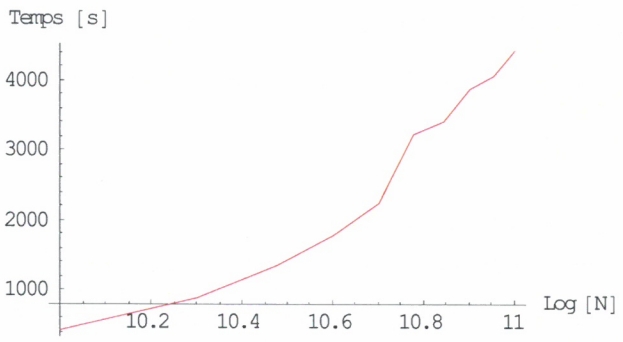


Figura 2.11

Pren uns temps d'execució força raonables i una memòria de $O\left(\sqrt{N}\right)$. Aquesta és utilitzada per executar el garbell d'Eratosthenes i extreure els primers en l'interval $\left[2, \sqrt{N}\right]$.

2.4 Garbell de Lagarias-Miller-Odlyzko

Pel Teorema dels Nombres primers quan $x \to \infty$ el valor de $\pi(x)$ és $\frac{x}{\ln x}$, per tant si s'utilitza el mètode més obvi per calcular $\pi(x)$, és a dir, comptant no es pot reduir la complexitat de l'algorisme de $O\left(\frac{x}{\ln x}\right)$. L'algorisme d'Eratosthenes, tot i la seva complexitat, ha estat durant molt temps l'única manera pràctica de calcular els primers $\leq x$. A finals del segle XIX l'astrònom alemany Meissel va descobrir que l'apropament combinatori modificant el garbell de Legendre podia produir un mètode més eficient; cal atribuir-li el mèrit de calcular $\pi\left(10^8\right)$ i $\pi\left(10^9\right)$ a mà tot i que en el darrer va donar un resultat massa petit per 56. Més tard, el 1959, Lehmer va executar un algorisme amb el mètode de Meissel simplificat en un IBM 701 per calcular el valor de $\pi\left(10^{10}\right)$ tot i que el seu resultat diferia en un del real.

Lagarias, Miller i Odlyzko el 1985 van adaptar el mètode de Meissel-Lehmer i van provar que es pot calcular $\pi(x)$ en $O\left(x^{2/3+\epsilon}\right)$ operacions i $O\left(x^{1/3}\log^2 x \log\log x\right)$ bits de memòria (veure [32]). Lagarias i Odlyzko van treballar en un altre algorisme basat en la integració numèrica de transformacions de la funció ζ de Riemann pel càlcul de $\pi(x)$ en $O\left(x^{1/2+\epsilon}\right)$ segons i $O\left(x^{1/4+\epsilon}\right)$ bits de memòria, però a nivell pràctic l'algorisme no pot competir per $x \leq 17$ amb el seu mètode anterior perquè les constants són massa grans (veure [33]). La millora de Deléglise i Rivat (1996) ha permès reduir el nombre d'operacions a $O\left(x^{2/3}\ln^{-2} x\right)$ però la memòria utilitzada a passat a ser $O\left(x^{1/3}\log^3 x \log\log x\right)$, (veure [52]).

No s'analitza ni s'implementa el mètode combinatori aquí mencionat per la seva complexitat, simplement era necessari mencionar que els garbells d'Eratosthenes, Atkin i Legendre no poden competir amb aquests per valors suficientment grans. El mètode de Déléglise i Rivat ha estat utilitzat per calcular $\pi\left(4\cdot 10^{22}\right) = 783\ 964\ 159\ 847\ 056\ 303\ 858$ en una xarxa d'ordinadors, amb un temps d'execució de 132 dies. Amb Eratosthenes, Atkin o Legendre aquest càlcul hagués estat impensable.

En una implementació a través de *Mathematica* 5.2, els temps d'execució han estat:

N	**Primers**	**Temps** (s)
10^1	4	0
10^2	25	0
10^3	168	0
10^4	1 229	0
10^5	9 592	0
10^6	78 498	0
10^7	664 576	0.15
10^8	5 761 455	0.16
$2*10^8$	11 078 937	0.16
$3*10^8$	16 252 325	0.16
$4*10^8$	21 336 326	0.16
$5*10^8$	26 355 867	0.16
$6*10^8$	31 324 703	0.16
$7*10^8$	36 252 931	0.16
$8*10^8$	41 146 179	0.16
$9*10^8$	46 009 215	0.16
10^9	50 847 534	0.19
10^{10}	455 052 511	0.219
10^{11}	4 118 054 813	0.953
10^{12}	37 607 912 018	4.641
10^{13}	346 065 536 839	23.359
10^{14}	3 204 941 750 802	113.75
$2*10^{14}$	6 270 424 651 315	249.969

Taula 2.12

L'algorisme és polinòmic en la magnitud de l'entrada i per tant exponencial en la longitud.

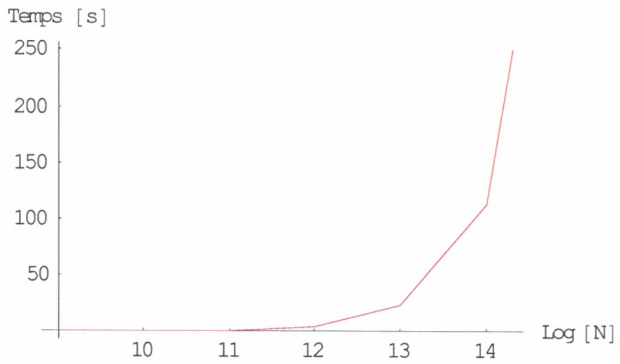


Figura 2.13

2.5 Bernhard Riemann i $\pi(x)$

Els matemàtics han intentat, fins al dia d'avui en va,
descobrir algun ordre en la seqüència dels nombres primers,
i tenim motius per pensar que aquest és un misteri
en el qual l'enteniment mai podrà penetrar.

Leonhard Euler
G. Simmons, *Calculus Gems*

En aquest subapartat es mostra com Georg Friedrich Bernhard Riemann, nascut a Breselenz el 17 de setembre de 1826 i mort a Selasca el 20 de juliol de 1866, va trobar una expressió que relacionava la funció Zeta amb la distribució dels nombres primers, (veure [48]). S'ha donat un apropament matemàtic sense provar estrictament totes les diferents expressions i s'investiga si la teoria matemàtica pot donar lloc a un algorisme de càlcul de $\pi(x)$.

La funció definida pels nombres complexos s basada en una sèrie Dirichlet infinita

$$\zeta(s) = \sum_{N=1}^{\infty} \frac{1}{N^s}$$

juntament amb la seva prolongació analítica rep el nom de funció Zeta de Riemann. La funció és convergent per tota $s > 1$. Està estretament vinculada amb la funció gamma però limitarem l'explicació al què ha permès avançar perquè en aquesta secció, més que en cap altre, hi ha el risc de perdre el fil.

La seva hipòtesi (o bé conjectura, els matemàtics no semblen haver-se aclarit en l'ús dels dos termes) més famosa diu que totes les arrels no trivials tenen com a part real un mig. Si bé Hardy va provar el 1914 que infinitat de zeros no trivials de la funció zeta de Riemann tenien com a part real un mig, tanmateix això no prova la hipòtesi del geni nascut a Breselenz el 17 de setembre de 1826 en el Regne de Hannover i que morí amb poc després del seu 40 aniversari possiblement d'una tuberculosi que patia des de la infància. "Infinitat de zeros" significa que de les infinites arrels que pugui tenir $\zeta(s)$, algunes d'elles tindran part real un mig, però no implica que totes compleixin la hipòtesi.

El 1859 Bernhard Riemannn va publicar un paper anomenat "Über die Anzhal der Primzahlen unter einer gegebenen Grösse", és a dir, *Sobre el nombre de nombres primers per sota una quantitat donada.* En aquest paper va relacionar el marge d'error de la funció $\pi(x)$ amb els zeros no trivials de la funció complexa que ell mateix va redefinir. Com passa en els $\mathbb{R}$, tota funció analítica es pot factoritzar i en $\mathbb{C}$ també es compleix. L'article és tremendament complicat i requereix uns coneixements matemàtics fora de l'abast, tanmateix he intentat comprendre part dels raonaments del geni que el van portar a calcular la funció de $\pi(x)$ de manera exacta i analítica.

La funció $\pi(N)$ denota el nombre de primers inferiors o iguals a N on $N \in \mathbb{N}$, però ara s'esten el domini de N als reals; això a primera vista no produeix cap canvi remarcable ja que $\pi(13) = \pi(13.29745) = \pi(13.9999) = 6$. La funció $\pi(x)$ pertany a aquella estranya classe de funcions amb salts, a l'hora de graficar simplement s'haurà de tenir en compte que quan $\pi(x) = 6$ quan x=13.2, 13.5, 13.9 però $\pi(x) = 5.5$ quan x=13, és una convenció.

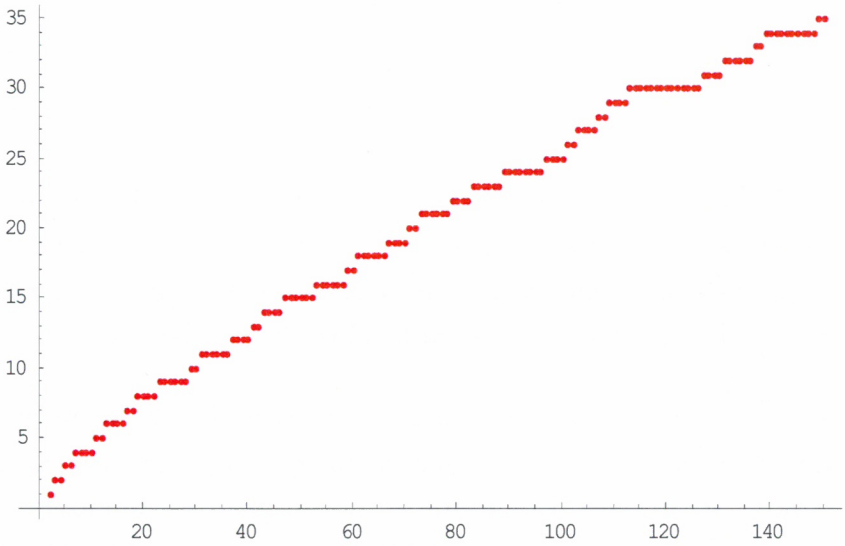


Figura 2.14

Riemann va introduir també una altra funció amb salts, el seu domini es situa en els naturals positius, es denota amb la J

$$J(x) = \pi(x) + \frac{1}{2}\pi\left(\sqrt[2]{x}\right) + \frac{1}{3}\pi\left(\sqrt[3]{x}\right) + \frac{1}{4}\pi\left(\sqrt[4]{x}\right) + \cdots$$

$$J(x) = \sum_{i} \frac{1}{i}\pi\left(\sqrt[i]{x}\right)$$

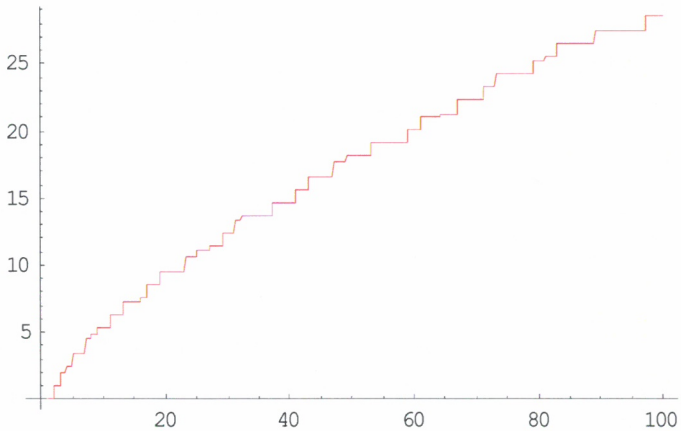


Figura 2.15

Si es té en compte el fet que $\pi(x)$ on $x \leq 2$ és igual a zero, es veu que la sèrie és sempre convergent. Si x=1000, a partir de la desena arrel $\pi(x)$ =0 perquè $\sqrt[10]{1000} = 1.995262315$. Per exemple, es pren x=50

$$J(50) = \pi(50) + \frac{1}{2}\pi(7.07) + \frac{1}{3}\pi(3.68) + \frac{1}{4}\pi(2.65) + \frac{1}{5}\pi(2.18)$$
$$15 + \frac{1}{2}\cdot 4 + \frac{1}{3}\cdot 2 + \frac{1}{4} + \frac{1}{5} = 18 + \frac{7}{60}$$

Al fer la gràfica de J(x) es veu que quan x és un primer la funció augmenta u, quan x és exactament un quadrat d'un nombre es produeix un salt d'un mig, quan és un cub salta un terç...

$\mu(N)$ denota la funció de Möbius. Rep el nom en honor del matemàtic i astrònom alemany August Ferdidnad Möbius (1790-1868). És molt utilitzada en teoria de nombres i juga un paper important en el paper de Riemann. La definició es troba en el punt 2.3.2.

En matemàtiques moltes vegades ens interessa expressar una funció en termes d'una altra i per tant s'han desenvolupat nombrosos instruments per invertir les relacions. Quan s'aplica la inversió de Möbius per expressar $\pi(x)$ en funció de J(x), desapareixeran tots els factors quadràtics i canviarà el símbol a negatiu per tots els primers o aquells que tinguin un nombre imparell de factors primers.

$$\pi(x) = J(x) - \frac{1}{2}J\left(\sqrt{x}\right) - \frac{1}{3}J\left(\sqrt[3]{x}\right) - \frac{1}{5}J\left(\sqrt[5]{x}\right) + \cdots$$
$$\pi(x) = \sum_{i=1} \frac{\mu(i)}{i} J\left(\sqrt[i]{x}\right)$$

Aquesta inversió és molt positiva perquè Riemann va trobar una manera de relacionar la funció J(x) i $\zeta(x)$. Cal remarcar que aquesta sumatòria és convergent perquè la funció J(x) és igual a zero per tota x menor a dos.

$$\pi(50) = J(50) - \frac{1}{2}J(7.07) - \frac{1}{3}J(3.6) - \frac{1}{5}J(2.1)$$

Impressionant,

$$18 + \frac{7}{60} - \left(\frac{1}{2}\cdot 4.5\right) - \left(\frac{1}{3}\cdot 2\right) - \frac{1}{5} = 15$$

Aquest és el nombre de primers inferiors o igual a 50!

Quan s'han vist demostracions de la infinitud de nombres primers ja havia aparegut la idea que la funció $\zeta(x)$ podia ser rescrita com el garbell d'Eratosthenes.

$$\zeta(s) = \frac{1}{\left(1-\frac{1}{2^s}\right)} \times \frac{1}{\left(1-\frac{1}{3^s}\right)} \times \frac{1}{\left(1-\frac{1}{5^s}\right)} \times \frac{1}{\left(1-\frac{1}{7^s}\right)} \times \frac{1}{\left(1-\frac{1}{11^s}\right)} \ldots.$$

Si s'aplica el logaritme en tots dos costats de la igualtat i es simplifiquen segons les seves lleis s'obté la sèrie:

$$\ln \zeta(s) = -\ln\left(1-\frac{1}{2^s}\right) - \ln\left(1-\frac{1}{3^s}\right) - \ln\left(1-\frac{1}{5^s}\right) - \ln\left(1-\frac{1}{7^s}\right) - \ln\left(1-\frac{1}{11^s}\right)$$

Sir Isaac Newton va aplicar sèries de Taylor de càlcul per desenvolupar un tipus similar d'expressió, on $x \in [-1,1]$:

$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + x^4$$

Si ara procedim a integrar ambdós costats de l'expressió, anant amb molta cura amb els signes s'obté:

$$-\ln(1-x) = x + \frac{x^2}{2} + \frac{x^3}{3} + \frac{x^4}{4} + \frac{x^5}{5} + \frac{x^6}{6} + \ldots.$$

canviant el signe de costat:

$$\ln(1-x) = x - \frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{4} - \frac{x^5}{5} - \frac{x^6}{6} - \ldots.$$

Retornant a la funció Zeta, es pot aplicar la idea a cada terme i expandre els logaritmes. Això sembla empitjorar, una sèrie infinita de sèries infinites....

$$\frac{1}{2^s}+\left(\frac{1}{2}\cdot\frac{1}{2^{2s}}\right)+\left(\frac{1}{3}\cdot\frac{1}{2^{3s}}\right)+\left(\frac{1}{4}\cdot\frac{1}{2^{4s}}\right)+\left(\frac{1}{5}\cdot\frac{1}{2^{5s}}\right)+\left(\frac{1}{6}\cdot\frac{1}{2^{6s}}\right)+....$$
$$\frac{1}{3^s}+\left(\frac{1}{2}\cdot\frac{1}{3^{2s}}\right)+\left(\frac{1}{3}\cdot\frac{1}{3^{3s}}\right)+\left(\frac{1}{4}\cdot\frac{1}{3^{4s}}\right)+\left(\frac{1}{5}\cdot\frac{1}{3^{5s}}\right)+\left(\frac{1}{6}\cdot\frac{1}{3^{6s}}\right)+....$$
$$\frac{1}{5^s}+\left(\frac{1}{2}\cdot\frac{1}{5^{2s}}\right)+\left(\frac{1}{3}\cdot\frac{1}{5^{3s}}\right)+\left(\frac{1}{4}\cdot\frac{1}{5^{4s}}\right)+\left(\frac{1}{5}\cdot\frac{1}{5^{5s}}\right)+\left(\frac{1}{6}\cdot\frac{1}{5^{6s}}\right)+....$$

Es pren un terme qualsevol de l'expressió de Newton, per exemple, $\left(3^{-1}\cdot 2^{-3s}\right)$ i es considera la funció x^{-s-1}, ara s'integra des de infinit a 2^3, i sent la integral convergent s'obté:

$$\int_{2^3}^{\infty} x^{-s-1}dx = -\frac{x^{-s}}{s}\Bigg]_{2^3}^{\infty} = 0-\left(-2^{3-s}\right)/s = \frac{2^{3-s}}{s}$$

Per tant l'expressió anterior $\left(\frac{1}{3}\cdot\frac{1}{2^{3s}}\right)$ pot ser rescrita segons una integral:

$$\frac{1}{3}\cdot\frac{1}{2^{3s}} = \frac{1}{3}\cdot s\cdot\int_{2^3}^{\infty} x^{-s-1}dx$$

La qüestió però, és què representa aquesta integral? Doncs és una àrea corresponent a la funció J(x), però no és el tipus d'àrea que generalment es veu sinó que està delimitada per dues rectes paral·leles a les abscisses, una a x=3^4 i l'altra a $x=\frac{1}{2^3}+\frac{1}{3}$, és a dir, d'alçada un quart.

L'àrea de la funció $J(x)$ està formada per la suma de franges que van de cada primer a infinit i d'alçada u; de cada quadrat de primer a infinit i d'alçada un mig; de la franja tancada per cada cub fins a infinit i d'alçada un terç.... i així progressivament. Evidentment l'àrea és infinita, de fet una sola d'aquestes franges ja té una àrea infinita, de manera que ens seria molt útil de fer que aquestes integrals convergissin.

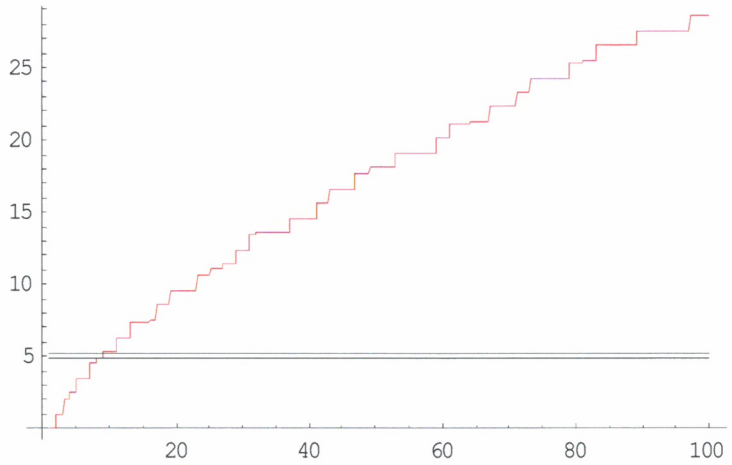


Figura 2.16

Es pot aconseguir aquest objectiu si es multiplica $J(x)$ per x^{-s-1} on x és major que 1; per exemple per x=50 $J(x)x^{-s-1} = 18.11667 \cdot 8 \cdot 10^{-6}$ on s val 2. Sembla que hi ha alguna possibilitat que l'àrea sigui finita, convergent; es buscant el valor de $\int_0^\infty J(x)x^{-s-1}dx$.

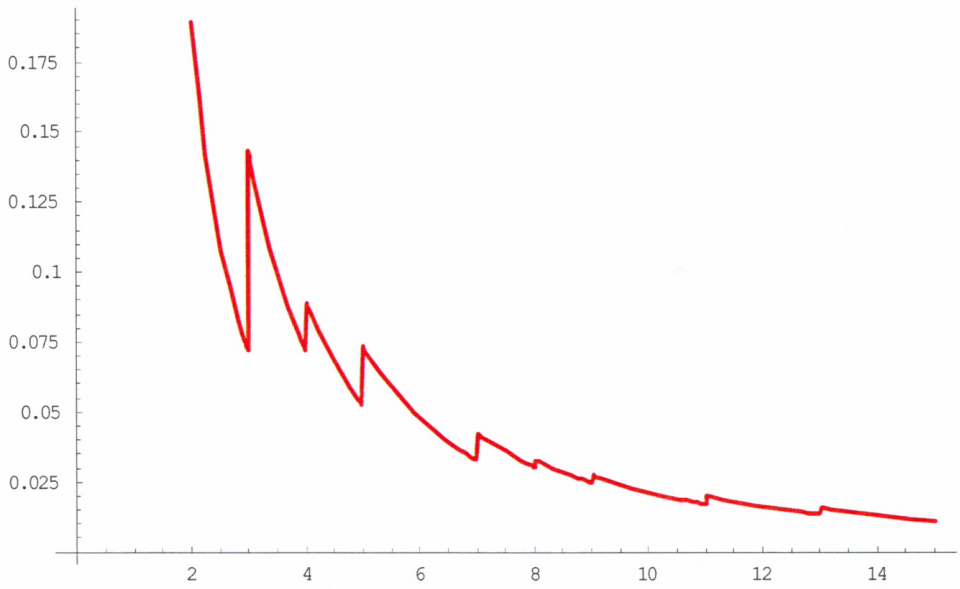


Figura 2.17

Per buscar aquesta àrea en realitat s'hauria de calcular la franja de cada primer fins a infinit, de cada quadrat de primer per un mig, de cada cub per un terç... i a més el nombre de primers és infinit de manera que es tropa amb una sumatòria infinita d'una sumatòria infinita d'integrals.

El que equival a:

$$\int_{2}^{\infty} 1 \cdot x^{-s-1} dx + \int_{2^2}^{\infty} \tfrac{1}{2} \cdot x^{-s-1} dx + \int_{2^3}^{\infty} \tfrac{1}{3} \cdot x^{-s-1} dx + \int_{2^4}^{\infty} \tfrac{1}{4} \cdot x^{-s-1} dx + \int_{2^5}^{\infty} \tfrac{1}{5} \cdot x^{-s-1} dx + \ldots$$

$$\int_{3}^{\infty} 1 \cdot x^{-s-1} dx + \int_{3^2}^{\infty} \tfrac{1}{2} \cdot x^{-s-1} dx + \int_{3^3}^{\infty} \tfrac{1}{3} \cdot x^{-s-1} dx + \int_{3^4}^{\infty} \tfrac{1}{4} \cdot x^{-s-1} dx + \int_{3^5}^{\infty} \tfrac{1}{5} \cdot x^{-s-1} dx + \ldots$$

....

és a dir, $$\sum_{p} \sum_{i} \int_{p^i}^{\infty} \tfrac{1}{i} \cdot x^{-s-1} dx$$

Si es repren el fet que un terme de la sumatòria de les sèries de Newton com $\left(4^{-1} \cdot 3^{-4s}\right)$ és igual a $\frac{1}{4} \cdot s \cdot \int_{3^4}^{\infty} x^{-s-1} dx$, al comparar l'expressió formada per la sumatòria infinita d'integrals amb l'expressió de Newton es veu que és s cops més gran. En realitat la sumatòria infinita d'integrals representa l'àrea sota la funció $J(x)x^{-s-1}$.

Per tant:

$$\frac{1}{s} \ln \zeta(s) = \int_{1}^{\infty} J(x) x^{-s-1} dx$$

Aquest és un dels resultats centrals de l'article de Riemann, va rescriure la formula del producte d'Euler,

$$\sum_{n} \left(n^{-s}\right) = \prod_{p} \left(1 - p^{-s}\right)^{-1}$$

$$\text{o bé } \zeta(s) = \prod_{p} \left(1 - p^{-s}\right)^{-1}$$

a través del càlcul; permetent així la seva millor manipulació.

De fet va obtenir una expressió directa de la funció $J(x)$ a través de la funció Zeta. En resum, va expressar $\pi(x)$ en funció de $J(x)$, i a la vegada $J(x)$ en

funció de $\zeta(s)$, de manera que podia calcular $\pi(x)$ a partir de $\zeta(s)$. A més, les propietats de la funció $\zeta(s)$ definien d'alguna manera les propietats de $\pi(x)$.

Aquesta definició de la funció $J(x)$ és una de les conclusions a les que arriba en el seu paper.

$$J(x) = \mathrm{Li}(x) - \sum_{\rho} \mathrm{Li}(x^{\rho}) - \ln(2) + \int_{x}^{\infty} \frac{dt}{t(t^2-1)\ln(t)}$$

En un primer cop d'ull sembla molt enrevessat però si s'analitza cada terme de manera aïllada. Primerament senyalar que $Li(x)$ té un valor que es pot determinar ràpidament amb una aplicació matemàtica potent com *Maple* 10 *o Mathematica* 5.2. El tercer terme és un logaritme neperià, cap problema: 0.69314718...

El darrer tot i semblar molt complicat és una integral impròpia però convergent per $x > 1$ i en el nostre cas no ens interessa cap cas amb $x < 2$; els seus valors varien entre 0.140010101143286926686917221371 (per x=2) i 0. De manera que els dos darrers termes, són quasi negligibles i oscil·len entre -0.69314718 i -0.553137079. En resum la part important de l'expressió és $\sum_{\rho} \mathrm{Li}(x^{\rho})$ on ρ són els zeros no trivials de la funció $\zeta(s)$, cal doncs ampliar el domini de $\mathrm{Li}(x^{\rho})$ als $\mathbb{C}$. Per *Mathematica* 5.2 cal redefinir ja que no executa bé l'ordre $\mathrm{Li}(x^{\rho})$ per nombres complexes.

$$\mathrm{Li}\left(x^{\frac{1}{2}+ir}\right) = \mathrm{ExpIntegralEi}\left[\left(\frac{1}{2}+ir\right)\ln[x]\right]$$

ρ representa cada parella d'arrels no trivials, una rere l'altra i la part irreal és sempre irracional. El primer zero no trivial és

$$\tfrac{1}{2} \pm 14.134725141734693790457251983562...\mathrm{i}$$
$$\tfrac{1}{2} \pm 21.022039638771554992628479593896...\mathrm{i}$$
$$\tfrac{1}{2} \pm 25.010857580145688763213790992562...\mathrm{i}$$
$$\tfrac{1}{2} \pm 30.424876125859513210311897530584...\mathrm{i}\,.$$
$$\tfrac{1}{2} \pm 32.935061587739189690662368964074...\mathrm{i}$$
$$\tfrac{1}{2} \pm 37.586178158825671257217763480705...\mathrm{i}$$
$$\tfrac{1}{2} \pm 40.918719012147495187398126914633...\mathrm{i}$$
$$...$$

Andrew Odlyzko ha calculat les primeres 100 parelles d'arrels no trivials amb 1000 decimals de precisió, cada arrel sigui $\frac{1}{2}+\mathrm{i}r$ té un conjugat $\frac{1}{2}-\mathrm{i}r$ i ha dissenyat l'algorisme més eficient i utilitzar per explorar regions remotament altes de l'eix imaginari.

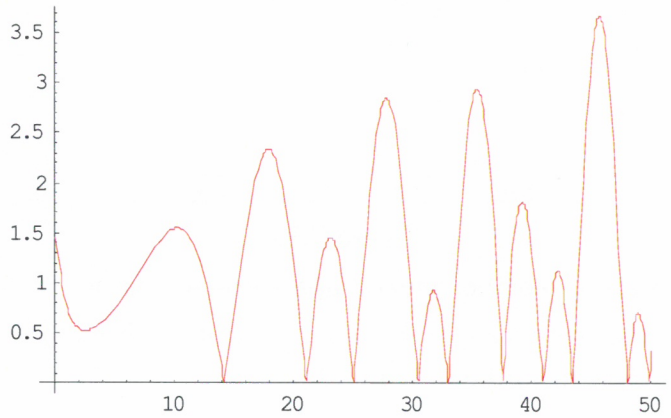


Figura 2.18

Aquest gràfic permet veure com es comporten els zeros de la funció Zeta, cada cop es troben més junts. El procés de recompte d'avaluació de $\sum_{\rho} Li\left(x^{\rho}\right)$ és complicat, perquè requereix moltes sumes tot i que la x sigui petita. L'ordre segons el què procedim per avaluar la suma és cabdal: si es vol assegurar la convergència de la part real cal començar pel zero més proper de l'eix dels reals i anar remontant l'eix imaginari, combinant cada vegada amb el conjungat complex. Quan es pren la integral logarítmica d'un nombre elevat a cada parella de zeros no trivials es veu que, l'expressió, va convergint a $\pm\pi\mathrm{i}$ amb un radi de $\sqrt{x}$.

En lloc de prendre la infinitat de zeros, deixant de banda que no es coneixen, es limita a un nombre fix com 200.000 per tal de simplificar la tasca. Aquest nombre serveix en principi d'aproximació per $\pi(x), x \leq 4 \cdot 10^{10}$. En general es necessiten aproximadament $\sqrt{x}$ zeros complexes per donar una aproximació prou bona com indica l'article: [47].

Riemann va aconseguir doncs escriure la funció $\pi(x)$ en termes d'una altra funció J(x), on J(x) estava escrita en termes de $\zeta(s)$. Per tant les propietats de $\pi(x)$ estan estretament lligades amb les de $\zeta(s)$ i si es coneixen les propietats de $\zeta(s)$ es pot saber molt més sobre la distribució dels nombres primers.

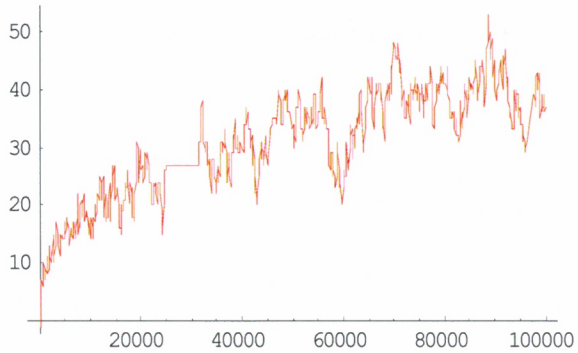


Figura 2.19

Aquest gràfic mostra el comportament aparentment caòtic de la funció $\pi(x)$ respecte la integral logarítmica que Riemann va aconseguir explicar a través de les arrels de la funció Zeta de variable complexa.

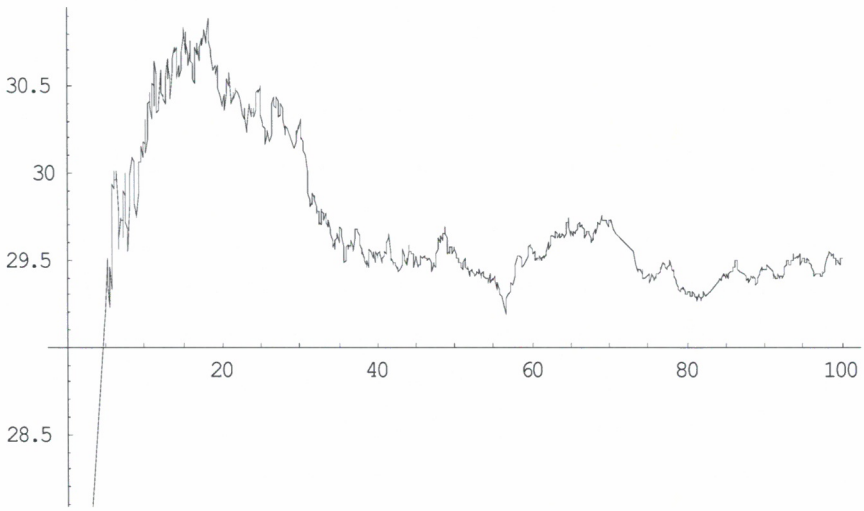


Figura 2.20

La obtenció dels valors que han permès la representació de la figura 2.20 ha requerit més de dues hores d'execució de *Mathematica* 5.2. La variable dependent en aquest cas és ρ, contada en milers; tot i que hi ha infinits zeros només se n'han considerat $2 \cdot 10^5$. Aquest gràfic mostra com $\sum_\rho Li\left(x^\rho\right)$ convergeix tot i que de manera molt lenta.

Codi per *Mathematica*

```
Plot[Sum[(ExpIntegralEi[(0.5+t[j]I)*Log[n]]+
ExpIntegralEi[(0.5-
t[j]I)*Log[n]]),{j,1,1000*x}],{x,1,100}]
```

On t[j] és un array que emmagatzema els valors dels zeros no trivials. Com es veu s' assumit la certesa de la hipòtesi de Riemann, tots els zeros tenen part real un mig.

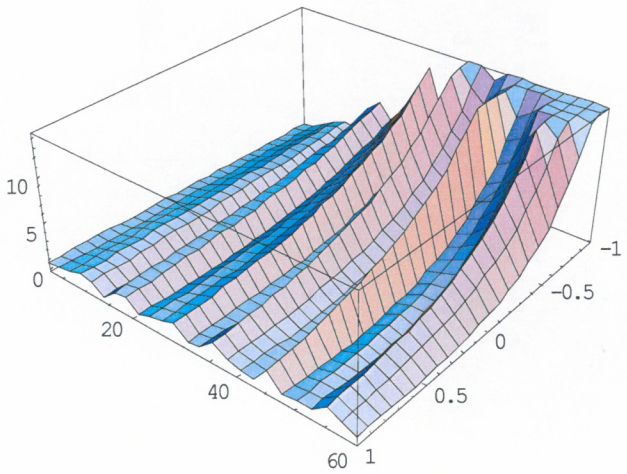


Figura 2.21

En la darrera gràfica apareixen representats els surcs que causen els zeros de la funció Zeta.

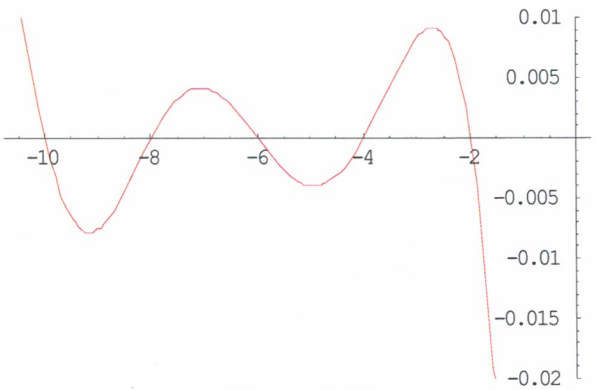


Figura 2.22

La figura 2.22 mostra els zeros trivials de la funció zeta pels reals, els enters parells negatius.

Codi per *Mathematica*

```
Plot3D[1/Abs[Zeta[x+I*y]^2],{x,-5,3.5},
{y,-50,50},PlotPoints->120,PlotRange->{0,15}
```

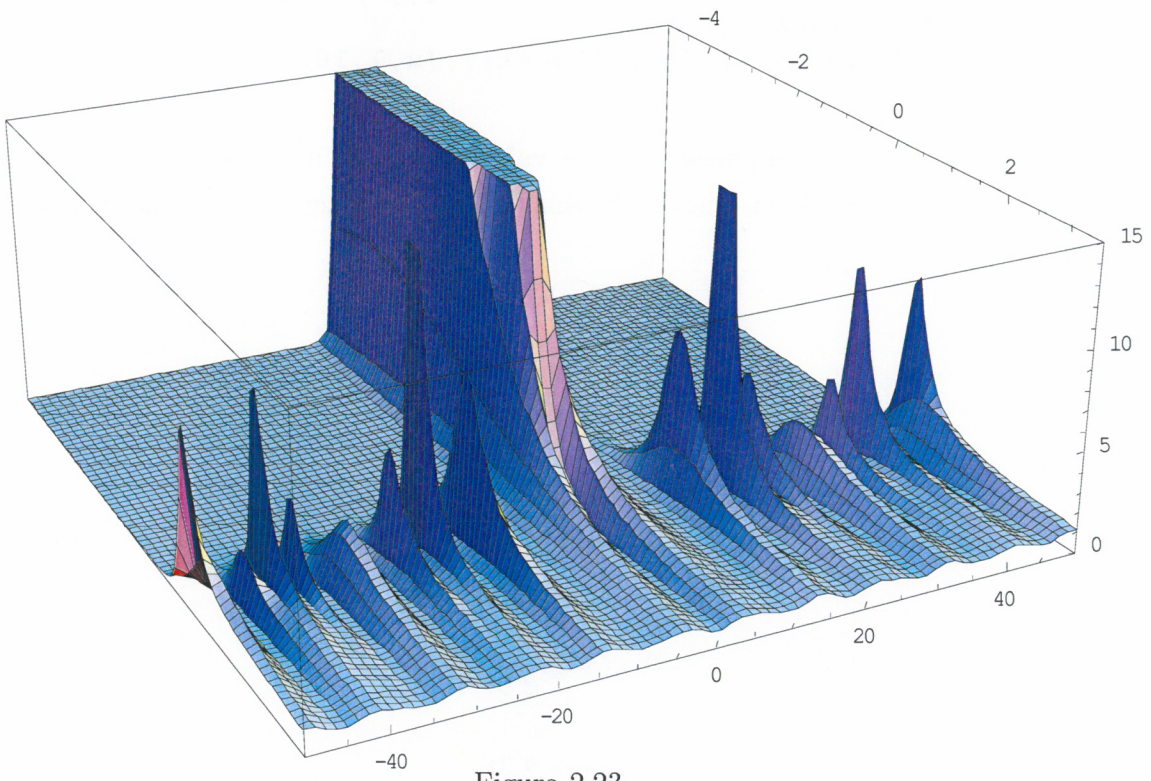


Figura 2.23

En aquest cas els zeros no trivials i els seus conjugats apareixen com a assímptotes de la gràfica.

$$N = 10^4$$

i	$\frac{\mu(i)}{i}\mathrm{Li}\left(N^{1/i}\right)$	$-\frac{\mu(i)}{i}\sum_{\rho}\mathrm{Li}\left(\left(N^{1/i}\right)^{\rho}\right)$	$-\frac{\mu(i)}{i}\ln 2$	$-\frac{\mu(i)}{i}\int_x^{\infty}\frac{dt}{t\left(t^2-1\right)\ln t}$
1	1246.1415	1.9002	-0.6931471	$5.161659129\cdot 10^{-10}$
2	-15.063115	-0.577681	0.3465741	$4.9378351314\cdot 10^{-6}$
3	-3.4714815	-0.30236	0.2310491	-0.0001023296215
5	-0.8785215	-0.0821036	0.1386291	-0.0011265120358
6	0.56777515	0.523602	-0.1155251	0.00204238534385
7	-0.39513915	0.0122924	0.09902115	-0.0031225733354
10	0.16802315	0.413473	-0.0693147	0.00672376373033
11	-0.13188415	0.0981064	0.06301341	-0.0079231019426
13	-0.083790815	0.12121	0.0533191	-0.0102199636604
	1226.853366	2.1067392	0.0536191	-0.0137283315
total	1229			

Taula 2.24

Aquest mètode a nivell pràctic no és viable perquè el segon terme és enormement difícil d'avaluar a causa de la seva convergència harmònica, és a dir de comportament assimptòtic logarítmic: ni sumant 100 000 termes s'assoleixen els tres decimals de precisió.

2. Conclusions

En aquesta secció s'ha pogut veure els esforços que els matemàtics han destinat al estudi de la funció $\pi(x)$, ja sigui des d'un punt de vista computacional basat en el càlcul o bé establint una formula matemàtica exacta. En certa manera és realment sorprenent que un mètode com el d'Eratosthenes, aparentment tant rudimentari, estigui, d'alguna manera, encara vigent ja sigui a través de les seves optimitzacions utilitzant formes quadràtiques irreductibles o bé simplement per la utilització de la idea de garbell que s'ha revel·lat absolutament fonamental.

El millor mètode pel càlcul de $\pi(x)$ que avui en dia existeix, com mostren els resultats experimentals, és el garbell de Miller-Lagarias-Odlyzko. Personalment el descobriment més sorprenent i enriquidor que he fet en aquest treball ha estat la funció Zeta, ζ, i el gran matemàtic alemany Riemann. Els seus resultats, tot i que des d'un punt de vista estrictament matemàtic són impecables, no són aplicables computacionalment perquè utilitza sèries de convergència molt lenta i que per tant tenen un error gens menyspreable.

"Ara, cinquanta anys després de la publicació de l'article de Riemann *Sobre la quantitat de nombres primer per sota d'un nombre donat*, tan sols hem començat a entendre i absorbir el que la creativitat i imaginació supremes de Riemann van produir. S'han fet progressos en el camí que Riemann va obrir sense por i que ha estat dubitatiu i lent; i només la famosa hipòtesi en la què es basa la seva tesi ha resistit tots els esforços de ser provada. "

E. Landau, 1909

3.Generació i verificació

El 317 és un primer, no perquè ho pensem, no perquè les nostres ments hagin estat modelades d'una manera o d'una altra, sinó perquè és així, perquè la realitat matemàtica està construïda d'aquesta manera.

Classe impartida per G. H. Hardy el 1915 a Cambridge

L'objectiu d'aquesta secció és analitzar la complexitat i el funcionament d'alguns dels tests de primeritat aleatoritzats més utilitzats, així com dur a terme una comparativa entre aquests tant a nivell pràctic com teòric. Per tant s'hauran d'implementar els algorismes i mostrar els resultats experimentals, és a dir, el comportament de cada mètode en el nostre *medi*: llenguatge, compilador i llibreries. Finalment s'analitzarà si aquests mètodes aleatoritzats són més competitius a nivell pràctic que els determinístics.

3.1 Tests de Monte Carlo

That I essentially am not in madness,
But mad in craft.
Hamlet Acte quart escena IV

Quan es busca obtenir una resposta a la pregunta: "N és primer o compost?" hi ha un conjunt d'algorismes de caràcter heurístic coneguts com mètodes de Monte Carlo que resolen aquest problema d'una manera estocàstica, no determinística, perquè per obtenir el resultat utilitzen una font atzarosa, millor dit pseudoaleatòria. Per conseqüent tenen un error simple i quan retornen el resultat *primer* existeix un marge d'error que s'ha de considerar. També convé senyalar que són altament eficients com es veurà tot seguit.

3.1.1 Test de Fermat

Definició 3.1.1.1 a tindrà un invers multiplicatiu en mòdul N només si $\text{mcd}(a,N)=1$, és a dir es pot trobar una k tal que $ak \equiv 1 (\text{mod } p)$.

Definició 3.1.1.2 La funció totient d'Euler que es denota com ϕ permet calcular la quantitat d'elements en el conjunt $[0..N-1]$ que són coprimers amb N i per tant invertibles en el mòdul. Per p primer $\phi(p)=p-1$ i $\phi(1)=1$.

Propietats 3.1.1.3

a) Per p i q primers, $\phi(p\cdot q)=(p-1)(q-1)$.

b) Si a, b $\in \mathbb{Z}_{\phi(N)}$ llavors $ab \bmod N \in \mathbb{Z}_{\phi(N)}$

c) Si a $\in \mathbb{Z}_{\phi(N)}$ si i només si existeix un b $\in \mathbb{Z}_N$ tal que $ab \bmod N = 1$

Prova

a) Sigui $N = p\cdot q$, el conjunt de residus seran $\mathbb{Z}_N = \{0,1,...,(pq-1)\}$. D'aquests hi haurà dos subconjunts no coprimers amb N, $\{p,2p,...,(q-1)p\}$ i $\{p,2p,...,(p-1)q\}$ a més del zero. Per tant:

$$\begin{aligned}\phi(N) &= pq-\left[(q-1)+(p-1)+1\right]\\ &= pq-(p+q)+1\\ &= (p-1)\cdot(q-1)=\phi(p)\phi(q)\end{aligned}$$

□

No es demostra absolutament totes les propietats perquè més que ajudar a seguir el raonament es podria perdre el fil conductor del capítol. No es preten provar tots els teoremes fonamentals de la Teoria dels Nombres sinó utilitzar-los per treure conclusions sobre uns algorismes.

Teorema d'Euler Si mcd(a,N)=1 llavors

$$a^{\phi(N)} \equiv 1 (\text{mod } N)$$

Prova

Considerant el conjunt $S = \left\{x_1, x_2, ..., x_{\phi(N)}\right\}$, si es multiplica cada element per a mòdul N s'obté: $S' = \left\{ax_1 \bmod N, ax_2 \bmod N, ..., ax_{\phi(N)} \bmod N\right\}$. Com que a és coprimer amb N i x_i és coprimer amb N, llavors ax_i ha de ser també coprimer amb N. Per tant tots els elements de S' són menors que N i coprimers amb ell. Si $ax_i \bmod N \equiv ax_j \bmod N$, llavors $x_i = x_j$ i per tant com que no hi ha duplicats en S':

$$\prod_{i=1}^{\phi(N)} (ax_i \bmod N) = \prod_{i=1}^{\phi(N)} x_i$$

$$\prod_{i=1}^{\phi(N)} ax_i \equiv \prod_{i=1}^{\phi(N)} x_i \bmod N$$

$$a^{\phi(N)} + \prod_{i=1}^{\phi(N)} x_i \equiv \prod_{i=1}^{\phi(N)} x_i \bmod N$$

$$a^{\phi(N)} \equiv 1 \pmod N$$

□

Teorema Petit de Fermat Sigui p primer i a un enter tal que $1 \leq a < p$, llavors es compleix:

$$a^{p-1} \equiv 1 \pmod p$$

Prova

Com que $\phi(p) = p - 1$ el Teorema Petit de Fermat és un cas particular del teorema d'Euler.

□

El Teorema Petit de Fermat té moltes aplicacions: per exemple pot ser utilitzat per trobar l'invers multiplicatiu de a mòdul p si dividim entre a als tots dos costats de la congruència s'obté:

$$a^{(p-1)-1} \equiv a^{p-2} \equiv a^{-1} \pmod p$$

Tot seguit es veurà que es pot aplicar aquest teorema per diferenciar els nombres primers dels compostos perquè quan $a^{N-1} \bmod N \neq 1$ llavors es té un certificat de la compositivitat de N. Aquest és un dels casos on s'ha d'aplicar l'exponciació per evitar els sobreeixements de memòria.

N	3	4	5	6	7	8	9	10	11	29	87	113	395	561
$2^{N-1}(\text{mod}\, N)$	1	0	1	2	1	0	4	2	1	1	4	1	174	1

Taula 3.1

Definició 3.1.1.6 Sigui a un enter tal que $1 \le a < N$ i $a^{N-1} \bmod N \neq 1$; s'anomena **testimoni** de compositivitat. És important fer observar que no representa cap pista sobre possibles factoritzacions.

La taula mostrava que el 561 és primer i això no és cert perquè $561 = 3 \cdot 11 \cdot 17$. Aquest fet permet introduir dos conceptes importants: la idea d'error i un tipus de nombre molt especial.

Definició 3.1.1.7 Sigui $1 \le a < N$, s'anomena **mentider** si $a^{N-1} \bmod N = 1$

Definició 3.1.1.8 S'anomena **nombre de Carmichael** tot compost imparell N que per totes les $a \in \mathbb{Z}_N^* = \mathbb{Z}_{\phi(N)}$. Tot nombre de Carmichael està format per tres o més factors primers. Gràcies als resultats obtinguts per W. R. Alford, A. Granville i C. Pomerance el 1994 se sap que n'hi ha infinits i que tenen una densitat assimptòtica de més de $x^{2/7}$.

Per comprendre el funcionament d'un possible algorisme basat en aquest mètode s'han d'introduir algunes propietats que caldrà demostrar.

Propietats 3.1.1.9 Sigui un enter $N > 1$,

a) Si $1 \le a < N$ satisfà $a^t \bmod N = 1$ per alguna $t \ge 1$ entera llavors $a \in \mathbb{Z}_{\phi(N)}$

b) Si per totes les a en el rang $1 \le a < N$ es compleix $a^{N-1} \bmod N = 1$ llavors N és primer.

Prova

a) Aquest és un cas especial de la propietat 3.1.1.3c; si $a^r \bmod N = 1$, llavors $r \cdot a^{r-1} \bmod N = 1$ per tant $a \in \mathbb{Z}_{\phi(N)}$.

b) Si per totes les a tal que $1 \le a < N$, $a^{N-1} \bmod N = 1$ llavors el conjunt $\mathbb{Z}_{\phi(N)} = \{1, \ldots, n-1\} \to \#\mathbb{Z}_{\phi(N)} = \phi(N) = N - 1$ i aquesta és la definició de primer.

□

De la propietat 3.1.1.9b se'n deriva un fet important, per un N compost la quantitat de testimonis de la compositivitat serà exactament $N-1-\phi(N)$. El problema és que per alguns casos aquest conjunt és força petit. Sigui $N = pq$; prenent el cas de $N = 85 = 5 \cdot 17$ es veu com les consideracions teòriques s'apliquen.

Testimonis en $\mathbb{Z}^*_n$	2, 3, 6, 7, 8, 9, 11, 12, 14, 19, 22, 23, 24, 26, 27, 28, 29, 31, 32, 36, 37, 39, 41, 42, 43, 44, 46, 48, 49, 53, 54, 56, 57, 58, 59, 61, 62, 63, 66, 71, 73, 74, 76, 77, 78, 79, 82, 83
Mentiders	1, 4, 13, 16, 18, 21, 33, 38, 47, 52, 64, 67, 69, 72, 81, 84
Múltiples de 5	5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80
Múltiples de 17	17, 34, 51, 68

Taula 3.2

La taula mostra els 48 testimonis, els 16 mentiders i els 20 múltiples de 5 i 17; seria una bona estratègia prendre els valors de a de manera aleatòria. En aquest cas la probabilitat d'error és $\frac{16}{82}$, representada pels 16 valors de a que són mentiders, cal observar que hi ha $N-3$ tries possibles diferents.

Teorema 3.1.1.10 Per un enter compost $N \geq 3$, tal que al menys hi ha un testimoni de compositivitat en $\phi(N)$, la probabilitat que aparegui com a primer no és major que $\frac{1}{2}$.

Prova

Com que s'ha afirmat que existeix un testimoni de compositivitat llavors ja es pot afirmar que el conjunt M de mentiders és un subgrup de $\mathbb{Z}_{\phi(N)}$. Es pot aconseguir una millor aproximació que simplement $\#M < \phi(N)-1$. El nombre d'elements de M ha de ser un divisor de $\phi(N) < N-1$, doncs es pot establir que $\#M \leq (N-2)/2$ perquè l'ordre del grup divideix al subgrup. De manera que la probabilitat que un element pres en $\{2,\ldots,N-2\}$ sigui un mentider és:

$$\frac{\frac{N-2}{2}-2}{N-3} = \frac{N-6}{2(N-3)} = \frac{N-6}{2N-6} < \frac{1}{2}$$

□

Es presenta l'algorisme aleatoritzat fruit d'aquest anàlisi:

```
Algorisme 3.1.1.10 (de Fermat)
Entrada: enter senar n≥ 3, enter e> 1.
function Fermat:boolean;
Var a,c,j:integer;
01 BEGIN
02  for j:=1 to e do
03   begin
04    a:=random{2,...,n - 2};
05    c := a^(n-1) mod n;
06    if c ≠ 1 then begin Fermat:=false; break; end;
07   end;
    Fermat:=true;
08 END.
```

Amb k iteracions, suposant que es prenen k as diferents, es veu que es pot reduir la probabilitat d'error a 2^{-k}. Si l'algorisme retorna *fals* es pot estar segurs de la compositivitat, sinó la probabilitat d'error quedarà marcada per l'anteriorment mencionada cota –assumint que pel nombre compost existeixi un valor de a tal que $\text{mcd}(a,N)=1$-. Tanmateix és important recordar la definició 3.1.1.8 perquè se'n desprèn que per certs compostos tots els valors de a que pertanyen a $\mathbb{Z}_{\phi(N)}$ són mentiders.

Quan es tracta d'un nombre de Carmiachel, la probabilitat d'error és $\phi(N)/N$ i això s'apropa molt a 1. Per exemple: quan N=651693055693681 llavors $\phi(N)/N = 0.9999650684$. L'única manera de corregir l'error és definir una constant d'iteracions que s'ha de multiplicar pel factor primer més petit, però quan els factors tenen una desena de dígits el mètode ja és inviable.

En conclusió, degut a les clares deficiències teòriques i pràctiques del mètode aquí analitzat s'hauran d'utilitzar un altre tipus de test de primeritat.

3.1.2 Test de Miller-Rabin

Els nombres primers tenen unes propietats aritmètiques úniques que poden ser utilitzades per identificar-los d'entre els compostos. És així com funcionen la majoria de tests de primeritat.

Definició 3.1.2.1 S'anomena a una arrel quadrada de 1 mòdul N si $1 \leq a < N$ i $a^2 \equiv 1 (\mathrm{mod}\, N)$.

Propietat 3.1.2.2 Per tot N sempre hi ha 1 i N-1 com arrels quadrades de 1 mòdul N, s'anomenen trivials.

Prova:
Cal veure si els dos casos compleixen la igualtat,

$$(N-1)^2 \equiv (-1)^2 \equiv 1 (\mathrm{mod}\, N)$$

□

Lema 3.1.2.3 Si p és primer, i es compleix que $a^2 \equiv 1 (\mathrm{mod}\, p)$, a només pot ser una de les arrels trivials.

Prova:
Sigui $a^2 \equiv 1 (\mathrm{mod}\, p)$, és a dir, $a^2 - 1 (\mathrm{mod}\, p) = 0$, i a més aplicant identitats notables es veu que $(a+1)(a-1) \mathrm{mod}\, p = 0$. Si el residu és zero això vol dir que o bé $p \mid (a+1)$ o $p \mid (a-1)$; com que el rang de a està limitat entre $1 \leq a < p$ es pot veure que per que es compleixi l'anterior condició o bé $a = 1$ o $a = p-1$.

□

Aquestes consideracions aparentment tant elementals no tant sols permetran analitzar un test de primeritat sinó que, es poden aplicar per tractar un dels millors algorismes de factorització dels què disposem avui en dia.

Del lema 3.1.2.3 es desprèn una primera aplicació: si es troba una arrel quadrada de 1 mòdul N no trivial llavors es disposa d'un testimoni de la compositivitat d'aquest. Si es considera la descomposició en factors primers de N, $N = p_1 \cdots p_r$, com que per cada primer en té exactament dues, N tindrà 2^r arrels quadrades de 1 mòdul N. Ara bé, la probabilitat de que per un N amb pocs factors primers aconseguim trobar una arrel quadrada no trivial és remota, exactament $2^r / N - 2$.

Però si encara es recorda el Teorema Petit de Fermat en ment, la congruència $a^{N-1} \equiv 1 (\mathrm{mod}\, N)$, la es pot transformar si es considera que N serà sempre senar i

que per tant $N-1=u\cdot 2^k$ per algun u imparell. Inserint la nova expressió en l'exponenciació anterior es veu que:

$$a^{N-1}\equiv a^{u\cdot 2^k}\equiv\left(a^u\right)^{2^k}\equiv\left(a^u \bmod N\right)^{2^k} \bmod N$$

Per tant es podrà calcular $a^{N-1}\equiv 1(\bmod N)$ en $k+1$ passos intermedis si es forma la seqüència $b_0=a^2 \bmod N$ i $b_i=b_{i-1}^2 \bmod N$ fins que $i=k$. Cal estudiar els possibles valors obtinguts en aquesta seqüència de manera exhaustiva perquè és a partir d'aquí que es podrà formalitzar l'algorisme.

- Cas 1 : $b_0\in\{1,N-1\}$ per tant $b_1=b_i=b_k$ de manera que no s'obté informació rellevant. Es pot aturar a b_1 i dir que N és primer. Aquest fenòmen es produeix per la propietat 3.1.1.2.

- Cas 2 : $b_0\notin\{1,N-1\}$, però hi ha un valor $b_i=N-1$.

- Cas 3 : $b_0\neq 1$ i $b_k=1$. A més no hi ha cap valor $b_i=N-1$ per $1\leq i<k$. Es pot trobar el valor mínim $b_i=1$ i aturar-nos perquè el resultat ja no canviarà i a més se sap que $b_{i-1}\notin\{1,N-1\}$ per tant s'ha trobat un factor primer no trivial: queda provat que N és compost.

- Cas 4 : $b_k\neq 1$ per tant N és compost. Cal recordar que s'havia d'haver obtingut 1 pel Teorema Petit de Fermat

b_0	b_1	...	b_{i-1}	b_i	b_{i+1}	...	b_{k-1}	b_k	Cas	Primer?
1	1	...	1	1	1	...	1	1	1	Si
N-1	1	...	1	1	1	...	1	1	1	Si
#	#	...	#	N-1	1	...	1	1	2	Si
#	#	...	#	#	#	...	#	1	3	No
#	#	...	#	1	1	...	1	1	3	No
#	#	...	#	#	#	...	#	N-1	4	No
#	#	...	#	#	#	...	#	#	4	No

Taula 3.3
Mostra els canvis en la seqüència.

Definició 3.1.2.4 Sigui N un enter senar major que tres, tal que $N-1=u\cdot 2^k$ on u és un enter senar i $k\geq 1$. S'anomena **testimoni** tot nombre a tal que $1\leq a<N$ si $a^u \bmod N \neq 1$ i $a^{u2^i} \bmod N \neq 1$ per totes les i en l'interval $1\leq i<k$. Per contra, si N és compost i a no satisfà les anteriors congruències serà un **mentider**.

La història d'aquest algorisme es remonta als anys 60, quan el rus M. Artjuhov va publicar un article on suggeria l'anteriorment analitzada seqüència per estudiar la primalitat d'un nombre. 10 Anys més tard, Miller va utilitzar la idea de Artjuhov (veure[4]) i es va adonar que podia crear un algorisme determinístic i de temps polinòmic si assumia que la Hipòtesi Extesa de Riemann era certa (veure [39]). Aquest va demostrar que el primer testimoni havia d'aparèixer abans de $O\left(\ln^2 N\right)$ i el seu resultat va ser corroborat el 1990 quan Bach (veure [5]) va provar el que la cota superior era $2\ln^2 N$. Cal però ser conscients que no és un algorisme incondicional perquè reposa en la certesa d'una altra hipòtesi matemàtica.

Tanmateix el 1980 Rabin (veure [45]) ja havia pres la idea de Miller per donar lloc al que es coneix avui en dia com test de Miller-Rabin, de naturalesa aleatoritzada i molt eficient.

Lema 3.1.2.5 Si l'algorisme retorna *compost,* el resultat és de ben segur correcte.

Prova
L'anàlisi de la seqüència mostra com s'ha trobat una arrel quadrada no trivial o bé que a és un testimoni de la compositivitat.

□

Algorisme 3.1.2.6 Miller-Rabin;
Entrada: enter senar $N \geq 3$

```
Var a,b,j:integer;
01 BEGIN
02  Trobar u senar i k tal que n-1=u·2^k;
03  a:=random{2,...,N - 2};
04  b:=a^u(mod n);
05  if b ∈ {1, N - 1} then return true;
06  for j:=1 to k-1 do begin
07       b := b^2(mod N)
08       if (b=N-1)then return true;
09       if (b=1)then return false; end;
10  return false;
11 END.
```

La complexitat d'aquest algorisme reposa principalment en la complexitat de l'exponenciació modular a la vegada que aquesta depèn dels mètodes d'aritmètica de múltiple precisió utilitzats. El nombre d'operacions aritmètiques és $O(2\log N) \approx O(\log N)$. Si s'utilitzen mètodes elementals el nombre d'operacions binàries és $O(\log N)^3$ mentre que si s'utilitza FFT, o Karatsuba es pot reduir a $O(\log N)^2$ amb el primer i $O(\log N)^{1+\log_2 3} \simeq O(\log N)^{2,584}$

Lema 3.1.2.7 La probabilitat que l'algorisme retorni primer quan N és compost és 4^{-1}.

Aquest lema no és gens trivial, és difícil de provar i com que a través del teorema xinès del residu només s'arriba a establir una cota de 2^{-1} simplement es remet el lector a [11]. L'error es pot reduir amb poques iteracions i amb un cost computacional molt raonable. En el meu cas com que volia ser capaç de verificar la primeritat de nombres d'una desena de milers de dígits he pres 10 iteracions. Però s'executen log N iteracions, la complexitat de l'algorisme en relació a la magnitud de l'entrada continua sent logarítmica i amb un error de $4^{-\log N}$.

Lema 3.1.2.8 Si l'algorisme Miller Rabin retorna compost, llavors N ho és.

Prova
En part aquesta demostració és una reelaboració de les idees que han permès analitzar la seqüència; en la línia 04 la variable b s'inicialitza, i en la i-èssima iteració canviant de $b_{i-1} = a^{u2^{i-1}} \bmod N$ a $b_i = a^{u2^i} \bmod N$. Per que s'obtingui compost, *false*, es necessita que $b_i = 1$. Només hi ha dos casos en què això pot passar:

- Cas 1 : si es retorna compost significa que $b_i = 1$, per les prèvies execucions de les línies 08 i 09, i de la línia 04 se sap que la seqüència $b_0, \ldots, b_{i-1} \notin \{1, N-1\}$, de manera que $b_0, \ldots, b_{k-1} \notin N-1$ i per tant a és un testimoni de la compositivitat.

- Cas 2 : s'executa la línia 10, les iteracions prèvies no han detectat cap $b_i = N-1$, per tant $b_0, \ldots, b_{k-1} \notin N-1$ i N és compost.

3.1.2.9 Resultats experimentals

Les taules que s'adjunten a continuació són fruït de l'execució de l'algorisme de Miller-Rabin, implementat per Free Pascal. S'ha volgut determinar la qualitat de l'anàlisi de la complexitat així com de la implementació; per fer-ho s'ha assumit que en aquest compilador es poden executar 10^8 operacions binàries per segon de manera que es pogués estimar el nombre d'operacions reals dutes a terme per l'algoritme. Cal assenyalar que com que l'exponenciació modular depèn en gran mesura del nombre, -concretament de la seva representació binària-, els valors de la constant oscil·len però cal remarcar també la tendència a anar augmentat progressivament, això és un problema del compilador i de la gestió de la memòria que a mesura que se'n requereix més es va enlentint. Per cada nombre, el programa ha executat 10 iteracions de manera que:

$$k\left(\log_2 10^N\right)^c = T \cdot op$$

$$\left(N \log_2 10\right)^c = \frac{T \cdot op}{k}$$

$$c \ln\left(N \log_2 10\right) = \ln \frac{T \cdot op}{k}$$

$$c = \frac{\ln T + \ln op - \ln k}{\ln N + \ln\left(\frac{\ln 10}{\ln 2}\right)} = \frac{\ln T + \ln op - \ln k}{\ln N + \ln \ln 10 - \ln \ln 2}$$

On T denota el temps en segons, op les operacions binàries (10^8) i k el nombre d'iteracions. De fet és una aproximació perquè no introduïm el nombre sinó una potència que representi el seu nombre de dígits. la probabilitat d'error quan el programa resol que el nombre és un primer, en el aquest cas és $9.53 \cdot 10^{-7}$.

Els nombres primers que han estat introduïts per veure el comportament del test han estat obtinguts de la pàgina web http://primes.utm.edu/curios/. Els programes que permeten generar primers prenen un nombre imparell aleatori, i van sumant dos fins que en troben un que passi el test de primeritat. Aquests resultats previs són imprescindibles per estimar quant pot trigar un programa com aquest en trobar un primer de N dígits.

El primer de la portada té 3799 dígits i va ser obtingut després de 16 hores d'execució; de fet vam tenir sort perquè només en va mirar 190, quan segons el teorema dels nombres primers, la probabilitat que un nombre de 3799 dígits sigui primer és aproximadament 1 entre 11 431.

Digits	Constant	Temps
25	2.91876	0:0:0:04
50	2.6325	0:0:0:07
60	2.60922	0:0:0:10
70	2.56886	0:0:0:12
75	2.28551	0:0:0:03
80	2.53503	0:0:0:14
85	2.52002	0:0:0:15
91	2.54884	0:0:0:21
95	2.503	0:0:0:18
100	2.48088	0:0:0:18
125	2.46234	0:0:0:28
150	2.46645	0:0:0:45
175	2.43243	0:0:0:53
190	2.42573	0:0:0:62
200	2.42526	0:0:0:70
225	2.42824	0:0:0:95
253	2.41302	0:0:1:14
275	2.42667	0:0:1:53
290	2.42579	0:0:1:73
300	2.42746	0:0:1:90
350	2.43039	0:0:2:82
401	2.43713	0:0:4:12

Taula 3.4

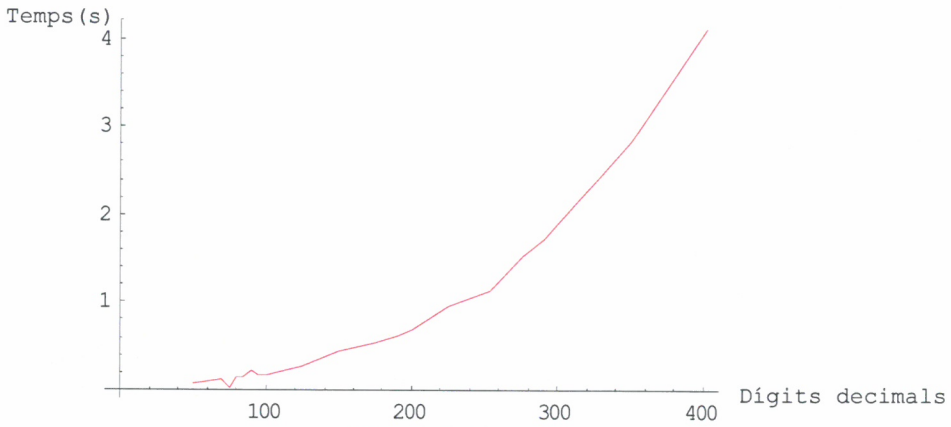


Figura 3.5

Aquesta gràfica representa els temps d'execució en segons, per nombres primers de fins a 400 dígits. De fet s'assembla força a aquesta funció $f(x) = 10(x \log_2 10)^{2.55} / 10^8$.

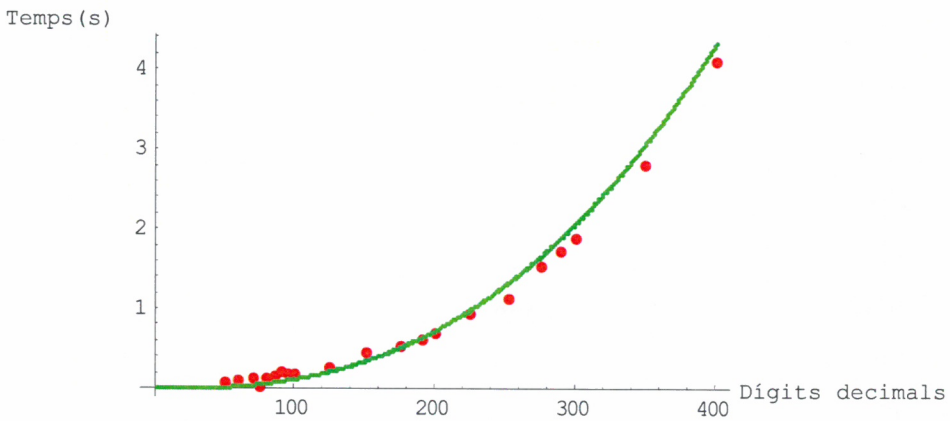


Figura 3.6

Tot seguit es pot observar les oscil·lacions de la constant que arriba un moment que s'estabilitzen. El gran surc es deu a l'aplicació de l'algorisme Karatsuba que redueix molt el temps d'execució pels valors en què resulta més eficient combinat amb el fet que aquest el nombre té una representació binària molt curta.

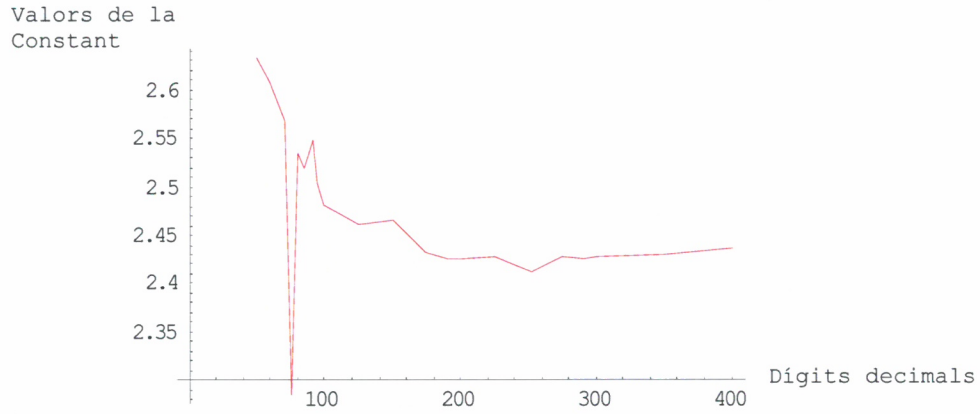


Figura 3.7

S'adjunten alguns valors més que permetran tenir una visió més aproximada del comportament assimptòtic de l'algorisme.

Digits	Constant	Temps
507	2.46386	0:0:8:90
601	2.44437	0:0:11:67
777	2.47554	0:0:27:93
1000	2.47556	0:0:52:17
1137	2.49842	0:1:26:54
1276	2.48808	0:1:45:89
1309	2.50384	0:2:8:76
1647	2.51482	0:4:11:53
1781	2.51806	0:5:14:95
2000	2.54093	0:8:35:84
2164	2.53083	0:9:36:12
2493	2.54408	0:15:28:93
2883	2.54484	0:22:33:96
2993	2.56015	0:28:34:65
3235	2.56055	0:35:0:20
3522	2.51663	0:28:50:29
3705	2.55673	0:47:47:39
3862	2.57233	1:1:35:23

Taula 3.8

Digits	Constant	Temps
3995	2.57204	1:7:0:64
5000	2.59074	2:23:6:98
16208	2.6225	68:24:53:19

Taula 3.9

Aquests són els valors de la constant. Si seguim augmentant la longitud de l'entrada queda palesa clarament la tendència a anar creixent, però cal senyalar que això es deu a la gran quantitat de memòria que el compilador ha de gestionar; el moment en el què resulta menys eficient, és molt probable que ja no sigui capaç de dur a terme les 10^8 operacions binàries per segon perquè li costi desplaçar-se per la memòria.

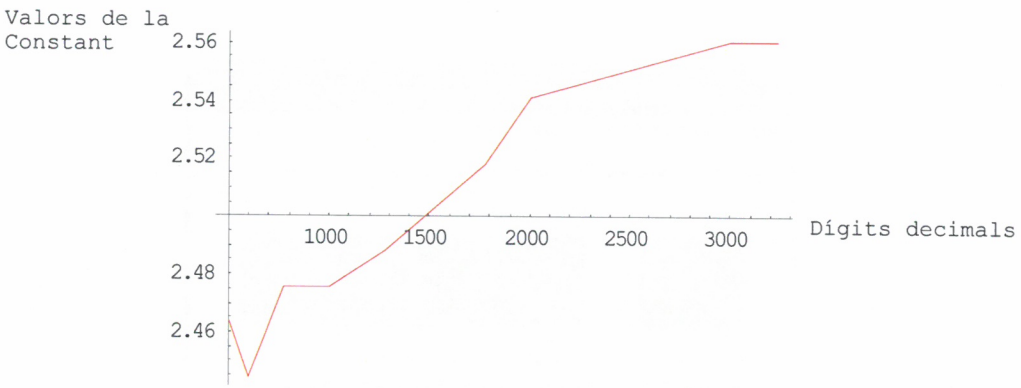


Figura 3.10

si es va més enllà, la tendència s'accentua i queda confirmada

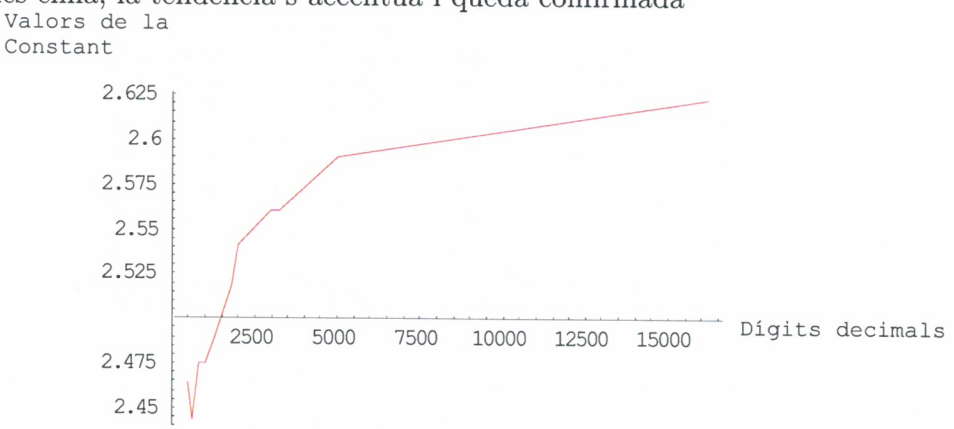


Figura 3.11

Pel que fa als temps d'execució la gràfica segueix cada cop més clarament la forma d'un polinomi de la forma: $f(x) = 10(x \log_2 10)^{2.55} / 10^8$

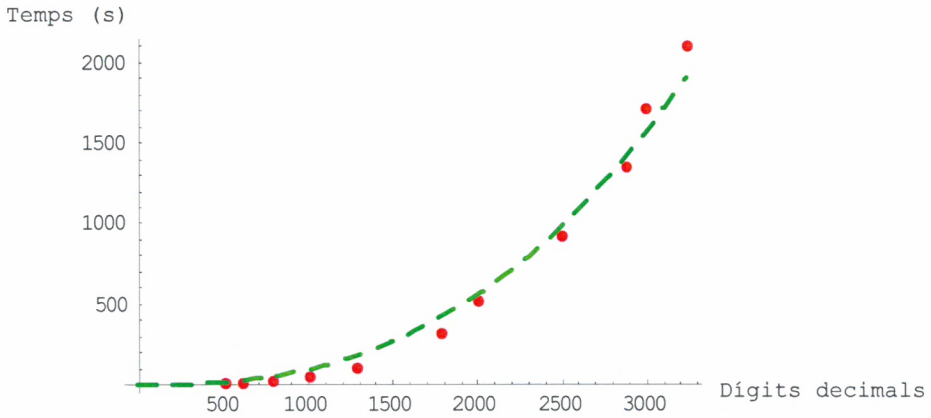


Figura 3.12

Si s'executa un Batch file, o arxiu de lots, es pot especificar la prioritat del procés i aconseguir un alt rendiment de processament

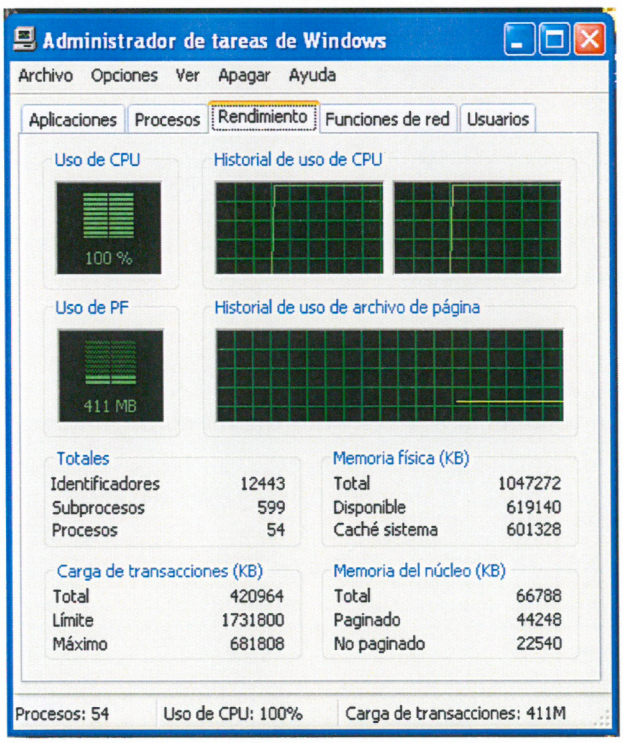


Figura 3.13

També és important aconseguir determinar quines aplicacions són estrictament necessàries pel funcionament del sistema operatiu i no executar cap de les altres. Per exemple si hi ha altres programes en execució, el temps real de processador dedicat al nostre programa és possible que només arribi al 50%.

3.1.3 Test de Solovay-Strassen

De fet el test de Solovay-Strassen és previ al de Miller-Rabin, ja que va ser presentat el 1976; les característiques generals de l'algorisme són força similars a les de Miller-Rabin perquè és aleatoritzat, de temps logarítmic en relació a la magnitud de l'entrada i d'error simple. Tanmateix mentre el funcionament de l'altre es basava en les arrels quadrades de 1 mòdul N, aquest depèn en els residus quadràtics.

Definició 3.1.3.1 Sigui $N \geq 2$, $a \in \mathbb{Z}$ i mcd(a,N)=1; es diu que a és **un residu quadràtic** mòdul N si $a \equiv x^2 (\mathrm{mod}\, N)$ per alguna $x \in \mathbb{Z}$. Contràriament si mcd(a,n)=1 però no és un residu quadràtic es diru que és un residu no-quadràtic.

Ser o no ser un residu quadràtic és una propietat de la congruència en què es calcula. A més els nombres que mcd(a,n)$\neq$ 1 ni se'ls considera, per tant en general es limita l'atenció al conjunt $\mathbb{Z}_N^*$.

Exemple 3.1.3.2 Si es pren $N = 17$, els quadrats mòdul N de $\{1,...,N-1\}$ són: 1, 4, 9, 16, 8, 2, 15, 13, 13, 15, 2, 8, 16, 9, 4, 1 i els residus quadràtics: 1, 2, 4, 8, 9, 13, 15, 16. Per $N = 22$ els residus quadràtics són: 1, 3, 5, 9, 15 i per $N = 35$ són: 1, 4, 9, 11, 16, 29.

Es pot realitzar una observació interessant per $N = 17$ hi ha 8 residus i 8 no-residus; es deu al fet que N és un primer.

Propietat 3.1.3.3 Elevant al quadrat els elements de $\mathbb{Z}_p^*$ s'obtinen com a molt $(p-1)/2$ valors diferents.

Prova

S'obtindran com a màxim $(p-1)/2$ residus diferents $(p-x)^2 \equiv x^2 \bmod p$. Els residus quadràtics $\{1,...,(p-1)/2\}$ seran diferents; imaginem que $y^2 \equiv x^2 (\mathrm{mod}\, p)$ per $1 \leq x \leq y < p/2$, llavors p divideix $y^2 - x^2 = (y+x)(y-x)$ i això només pot passar si $y = x$ o $y = p - x$

□

Lema 3.1.3.4 El Criteri d'Euler diu que va provar que si p és primer, el conjunt de residus quadràtics (RQ) té una mida de $(p-1)/2$; a més per $a \in \mathbb{Z}_p^*$,

$$a^{(p-1)/2} \begin{cases} \equiv 1 \pmod{p} & \text{si } a \text{ és un RQ mòdul } p \\ \equiv -1 \pmod{p} & \text{si } a \text{ no és RQ mòdul } p \end{cases}$$

Lema 3.1.3.5 Si p és un nombre primer amb $p \equiv 3 \pmod 4$, en aquest cas per tot a que sigui un residu quadràtic en $\mathbb{Z}_p^*$, l'element $x \equiv a^{(p-1)/4} \pmod p$ satisfarà $x^2 \equiv a \pmod p$.

Prova

Primerament cal senyalar que $p \equiv 3 \pmod 4$ perquè és l'única manera que $(p-1)/4 \in \mathbb{Z}$. Després la igualtat és immediata si es té en compte que:

$$a^{(p+1)/4} \equiv \left(a^2\right)^{(p+1)/4} \equiv a^{(p+1)/2} = a^{(p-1)/2+1} \equiv a^{(p-1)/2} a \equiv a \pmod p$$
$$a^{(p-1)/2} \equiv 1 \pmod p$$

□

Definició 3.1.3.6 Per un nombre primer $p \geq 3$ i un enter a s'anomena $\left(\frac{a}{p}\right)$ símbol de Legendre si compleix:

$$\left(\frac{a}{p}\right) = \begin{cases} 1, \text{ si } a \text{ és un RQ mòdul } p \\ -1, \text{ si } a \text{ no és RQ mòdul } p \\ 0, \text{ si } p \mid a \end{cases}$$

Les seves propietats estan lligades al criteri d'Euler

Propietats 3.1.2.7 Sigui $p \geq 3$

- $\left(\frac{a}{p}\right) = \left(\frac{a \bmod p}{p}\right)$ per tots els a
- $\left(\frac{a \cdot b}{p}\right) = \left(\frac{a}{p}\right) \cdot \left(\frac{b}{p}\right)$ per $a, b \in \mathbb{Z}$
- $\left(\frac{a \cdot b^2}{p}\right) = \left(\frac{a}{p}\right)$ per $a, b \in \mathbb{Z}$ i $b \bmod p \neq 0$.
- $\left(\frac{-1}{p}\right) = (-1)^{(p-1)/2}$, p-1 és un RQ si $p \equiv 1 \pmod p$.

Propietats 3.1.3.8 Sigui $N \geq 3$ un enter senar amb descomposició $N = p_1 \cdot p_2 \cdots p_r$, llavors $\left(\frac{a}{N}\right)$ s'anomena símbol de Jacobi de a i N i s'evalua així:

$$\left(\frac{a}{N}\right) = \left(\frac{a}{p_1}\right) \cdot \left(\frac{a}{p_2}\right) \cdots \left(\frac{a}{p_r}\right)$$

El Símbol de Jacobi és una generalització del Símbol de Legendre, en el sentit que permet verificar si a és un residu quadràtic mòdul n, sent n primer o no (sempre que el mcd(a,N)=1).

Propietats 3.1.3.9 Si l'enter $N \geq 3$ i a i b són també enters llavors:

a) $\left(\frac{1}{N}\right) = 1$ i $\left(\frac{0}{N}\right) = 0$

b) $\left(\frac{a \cdot b}{N}\right) = \left(\frac{a}{N}\right) \cdot \left(\frac{b}{N}\right)$, si mcd($a$,$b$)=1

c) $\left(\frac{2^{2k} \cdot a}{N}\right) = \left(\frac{a}{N}\right)$ i $\left(\frac{2^{2k+1} \cdot a}{N}\right) = \left(\frac{2}{N}\right) \cdot \left(\frac{a}{N}\right)$ per $k \geq 1$

d) $\left(\frac{a}{N}\right) = \left(\frac{a \bmod N}{N}\right)$ per a qualsevol valor de a

Hi ha moltes més propietats que poden ser demostrades però l'enfocament es limita a només algunes d'elles. Al final de la secció aquesta idea prendrà sentit.

Llei de Reciprocitat Quadràtica Siguin $M, N > 2$ enters senars,

$$\left(\frac{M}{N}\right) \begin{cases} \left(\frac{N}{M}\right), & \text{si } N \text{ o } M \equiv 1 \pmod 4 \\ -\left(\frac{N}{M}\right), & \text{si } N \text{ o } M \equiv 3 \pmod 4 \end{cases}$$

Propietats 3.1.3.11 Si N¿2 és un enter senar,

$$\left(\frac{2}{N}\right) \begin{cases} 1, & \text{si } N \equiv 1 \text{ o } N \equiv 7 \pmod 8 \\ -1, & \text{si } N \equiv 3 \text{ o } N \equiv 5 \pmod 8 \end{cases}$$

Si es fa ara un breu repàs de les identitats que s'ha vist, es pot constatar que l'avaluació de $\left(\frac{a}{N}\right)$ es pot realitzar de la següent manera:

1. Si a=0, el resultat és 0
2. Si a=1, el resultat és 1
3. Si a¿N llavors el resultat és $\left(\frac{a \bmod N}{N}\right)$
4. Si (a¿2) i a o $N \equiv 1 \pmod 4$, el resultat és $\left(\frac{N \bmod a}{a}\right)$
5. Si (a¿2) i a i $N \equiv 3 \pmod 4$, el resultat és $\left(\frac{-N \bmod a}{a}\right)$
6. Si a és múltiple de 4, el resultat és $\left(\frac{a/4}{N}\right)$
7. Si $2|a$, el resultat és $\left(\frac{-a/2}{N}\right)$ si $N \pmod 8 \in \{3,5\}$ i $\left(\frac{a/2}{N}\right)$ si $N \pmod 8 \in \{1,7\}$.

Tenint en compte aquestes propietats ja es pot dissenyar un algorisme recursiu per avaluar el Símbol de Jacobi.

```
function 3.1.3.12 Jc(a,n :longint):longint;
 BEGIN
  if n=0 then Jc:=1
         else if a=0
              then Jc:=0
              else if (a mod 2 =1)
                       then if (a mod 4=3)and(n mod 4=3)
                                then Jc:=-Jc(n mod a, a)
                                else Jc:=Jc(n mod a,a)
                       else if (n mod 8=3)or(n mod 8=5)
                                then Jc:=-Jc(a div 2,n)
                                else Jc:=Jc(a div 2,n);

 END;
```

Però donat que es vol assegurar l'eficiència de l'algorisme, és preferible un procediment iteratiu que no carregui en excés la memòria de l'ordinador. Es tornen a aplicar les propietats que s'han anat veient, i es reserva una variable al símbol.

```
Algorisme 3.1.3.13
Fucntion Jc(a,n :longint):longint;
var j, i, s, x : integer;
01 BEGIN
02 j:=a mod n;
03 i:=n; s:=1;
04 while (j>1) do
05  begin
06   while (j mod 4=0) do j:= j div 4
07   if (j mod 2=0)then
08    begin
09     if (i mod 8 in [3,5]) then s:=-s;
10     j:=j div 2;
11    end;
12   if (j=1) then break;
13   if (j mod 4)=(i mod 4)=3 then s:=-s;
14   x:=j; j:= i mod j; i:=x;
15  end;
16 Jc:=s*j;
17 END;
```

Teorema 3.1.2.14 L'algorisme 3.1.1.13 funciona de manera correcta, és a dir:

$$s\left(\frac{i}{j}\right)=\left(\frac{a}{N}\right)$$

Prova

A l'inici el cas és trivial perquè els valors són els mateixos; en la línia 06 es divideix entre quatre però això per la propietat 3.1.3.9.c no altera el resultat. En la línia 10 es divideix entre dos però això tampoc altera el resultat perquè la línia anterior s'ha multiplicat s per $\left(\frac{2}{j}\right)$. Si s'arriba a la línia 13 segur que i és un senar. A la línia 13 es canvia el valor de s si cal i a la 14 es canvia $\left(\frac{i}{j}\right)$ per $\left(\frac{j \bmod i}{i}\right)$, això tampoc altera el resultat per la propietat 3.1.3.9.d.

□

Teorema 3.1.3.15 L'algorisme retorna el resultat en $O(\log N)$ operacions aritmètiques.

Prova

L'anàlisi de l'algorisme en realitat és equivalent a trobar el nombre màxim d'iteracions de les línies 04-15; s'haurà d'estudiar la seqüència de (i_0, j_0), $(i_1, j_1),\ldots,(i_k, j_k)$ i es veu que quan s'inicia la línia 04 en la k-iteració $j_k \geq 2i_{k+2}$. Per tant el nombre d'iteracions serà com a molt $2\lceil \log_2 N \rceil = O(\log N)$.

□

Cal concretar en un mètode pràctic totes aquestes propietats matemàtiques sense les quals no s'hagués entès el veritable funcionament de l'algorisme. L'algorisme de Solovay-Strassen és un test de primeritat aleatoritzat desenvolupat per Robert Solovay i Volker Strassen l'any 1976 (veure, [58]). La idea en què es basa l'algorisme és la següent: si N és primer llavors pel criteri d'Euler, lema 3.1.3.4, $\left(\frac{a}{N}\right)$ i $a^{(N-1)/2} \bmod N$ han de ser iguals per totes les $1 \leq a < N$.

Lema 3.1.3.16 Si un primer p cumpleix $p \geq 3$ llavors

$$\left(\frac{a}{N}\right) \cdot a^{(N-1)/2} \equiv 1 \pmod{N} \text{ per } 1 \leq a < N$$

Definició 3.1.3.17 Si N és un nombre compost i per un valor de a tal que $1 \leq a < N$, es compleix $\left(\frac{a}{N}\right) \cdot a^{(N-1)/2} \equiv 1 \pmod{N}$, es diu que a és un mentider. Si no ho complís llavors seria un testimoni de la compositivitat.

Lema 3.1.3.18 Sigui $N \geq 3$, llavors tot mentider de la definició 3.1.3.17 també serà un mentider de Fermat.

Prova

Com que a és un mentider, per 3.1.3.17 $\left(\frac{a}{N}\right) \cdot a^{(N-1)/2} \equiv 1 \pmod{N}$, per tant

$\left(\frac{a}{N}\right) \in \{1,-1\}$ i $1 = \left(a^{(N-1)/2}\left(\frac{a}{N}\right)\right)^2 \bmod N = a^{N-1} \pmod{N}$

□

Lema 3.1.3.19 El nombre d'elements de $\mathbb{Z}_N^*$ que són mentiders no pot ser major que $\frac{1}{2}\phi(N)$.

Prova

Cal remetre al lector a la abundant bibliografia perquè la prova no és immediata i tampoc té un interès especial per als objectius d'aquest treball. Quan N és primer no hi ha possibilitat d'error, però quan és compost la probabilitat que es prengui un mentider és 1/2, de manera que després de k iteracions es pot afirmar que el marge d'error serà de 2^{-k}.

```
Algorisme 3.1.3.20
Function Solovay-Strassen(n, k :longint):boolean;
var j : integer;

01 BEGIN
02  for j:=1 to k do
03   begin
04    a:=random {2,...,N - 2}
05    if (a/N) · (a^((N-1)/2) mod N) ≠ 1 then Solovay-Strassen:=false; break;
06   end;
07  Solovay-Strassen:=true;
08 END;
```

Lema 3.1.3.21 La complexitat de l'algorisme és $O\left(\log^3 N\right)$ per mètodes elementals i $O\left(\log^2 N\right)$ si s'utilitzen subrutines de multiplicació molt eficients.

Prova

Per cada iteració el que té cost computacional serà l'avaluació del símbol de Jacobi i l'exponenciació; per tant es veu

$$O\left(\log N\right) + O\left(\log^c N\right) \approx O\left(\log^c N\right)$$

per $c \geq 1$.

3.1.3.22 Resultats experimentals

Les mateixes consideracions experimentals de l'apartat 3.1.1 són aquí aplicables tanmateix cal veure que en aquest algorisme quan retorna primer, la probabilitat d'error és un mig i en el nostra cas, després de 10 iteracions, $9.765 \cdot 10^{-4}$. Hi ha múltiples factors que afecten el valor de la constant per aquest motiu va canviant, tanmateix ha de quedar clar que en principi té un valor constant.

Dígits	Resultat	Temps
25	3.10225	0:0:0:09
50	2.80605	0:0:0:17
60	2.74012	0:0:0:20
70	2.67156	0:0:0:21
75	2.75493	0:0:0:40
80	2.62396	0:0:0:23
85	2.59577	0:0:0:23
91	2.61703	0:0:0:31
95	2.65433	0:0:0:43
100	2.56303	0:0:0:29
127	2.51488	0:0:0:40
155	2.49957	0:0:0:60
175	2.45707	0:0:0:62
191	2.41607	0:0:0:59
200	2.44772	0:0:0:81
225	2.44902	0:0:1:09
253	2.4267	0:0:1:25
271	2.44738	0:0:1:70
289	2.42951	0:0:1:76
300	2.46778	0:0:2:51
355	2.44205	0:0:3:17
401	2.46292	0:0:4:96

Taula 3.14

Per valors petits és poc eficient però a mesura que la longitud de N és major el fet d'avaluar el símbol de Jacobi perd importància.

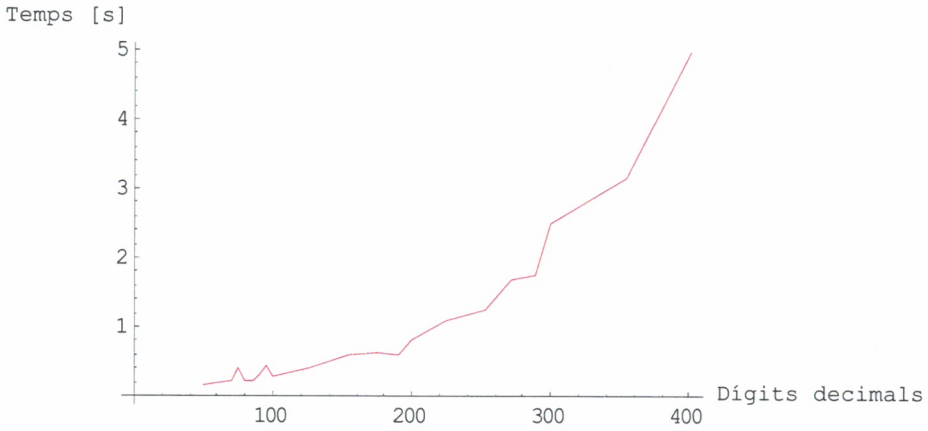


Figura 3.15

La semblança amb la funció $f(x) = 10(x \log_2 10)^{2.44} / 10^8$ s'anirà confirmant a mesura que afegim més dades.

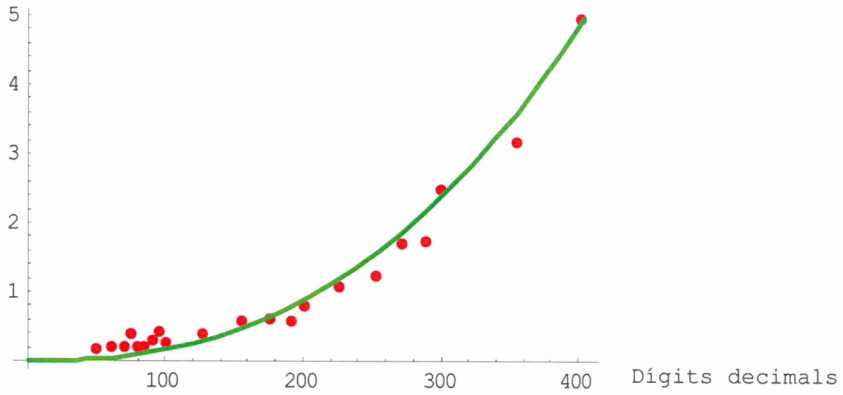


Figura 3.16

La constant que s'està intentat determinar té el comportament següent:

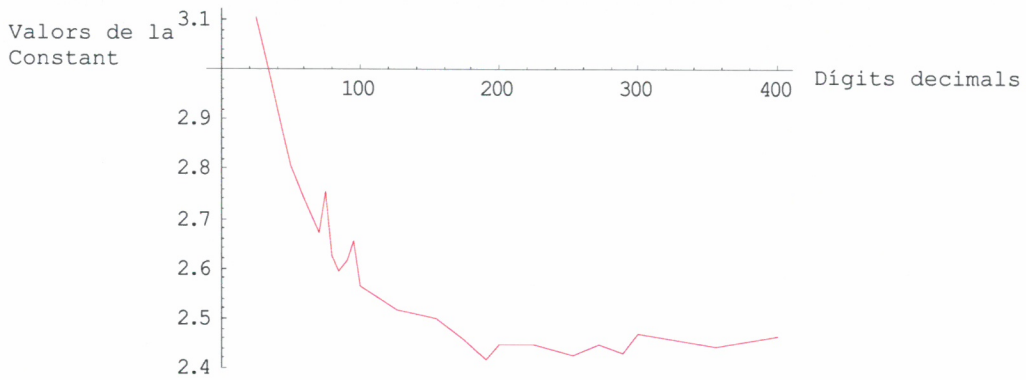


Figura 3.17

Dígits	Resultat	Temps
507	2.45479	0:0:8:32
601	2.44639	0:0:11:85
777	2.45933	0:0:24:59
1000	2.47473	0:0:51:82
1276	2.48911	0:1:46:81
1505	2.51038	0:3:13:07
1781	2.51829	0:5:15:60
2000	2.53726	0:8:19:45
2506	2.5991	0:25:46:80
2993	2.59356	0:26:47:48
3235	2.57594	0:40:22:65
4332	2.58225	1:31:00:07
5000	2.58884	2:20:30:32
5941	2.56782	2:58:21:45
6002	2.58207	3:30:50:48
6533	2.58551	4:31:36:17
7054	2.58815	5:40:5:62

Taula 3.18

A partir de l'anterior taula es pot confeccionar la següent gràfica que confirma els resultats teòrics i el que els primers càlculs havien deixat entreveure, es pot verificar la primeritat en temps polinòmic en funció de la longitud de l'entrada.

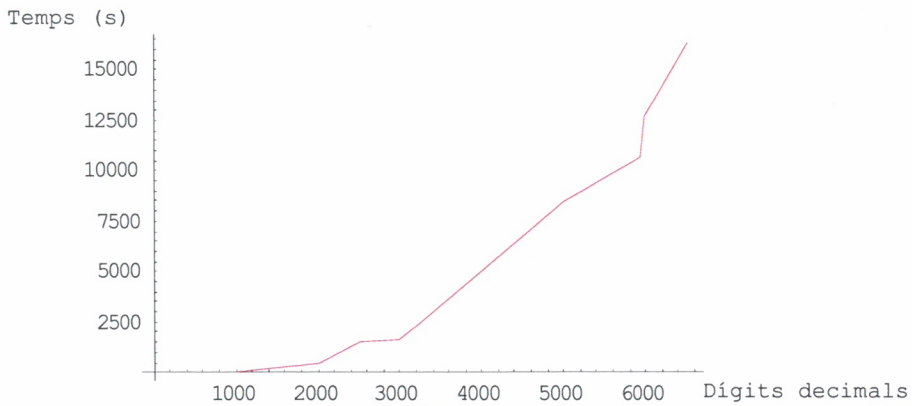


Figura 3.19

La semblança amb la funció $f(x) = 10(x \log_2 10)^{2.5849} / 10^8$, que és similar a la funció $g(x) = 10(x \log_2 10)^{1+\log_2 3} / 10^8$, és força clara. Si es consideren les dues gràfiques sobre el mateix eix de coordenades, la similitud queda confirmada.

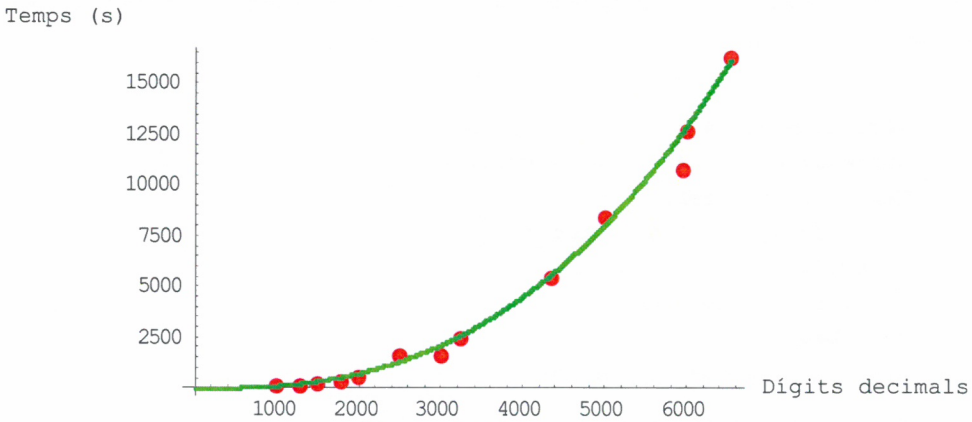


Figura 3.20

La constant poc a poc es va estabilitzant com mostra la gràfica següent.

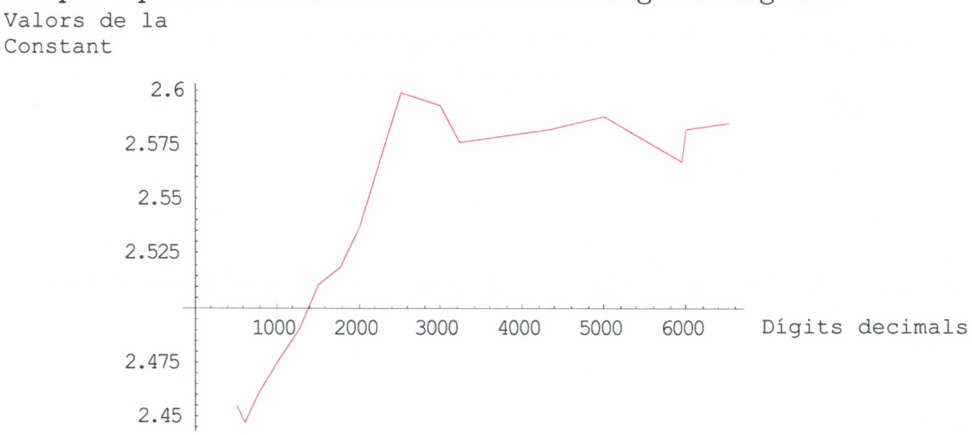


Figura 3.21

3.1.4 Test de Lehmann

L'algorisme que es descriu a continuació va ser publicat per D. J. Lehmann el 1982, pot ser interpretat com una revisió del test de Solovay-Strassen perquè utilitza els residus quadràtics, però amb la diferència que no requereix l'avaluació del símbol de Jacobi.

```
Algorisme 3.1.4.1 Lehmann:boolean;
Entrada: enter senar N≥ 3, enter k> 1.
Var a,c,j:integer;
    b:array [1..k]of integer;
01 BEGIN
02  for j:=1 to k do
03   begin
04    a:=random{1,...,N − 1};
05    c := a^((N−1)/2) mod N;
06    if c ∉ {1, N − 1} then begin Lehmann:=false; break; end;
07                   else b[j]:=c;
08   end;
09  if b[1]=···=b[k]=  1 then Lehmann:=false;
10                       else Lehmann:=true;
11 END.
```

Teorema 3.1.4.2 La probabilitat que el resultat erroni sigui retornat després de k iteracions és de 2^{-k}.

Prova

Si es repren el Criteri d'Euler (Lema 3.1.3.4) es veurà que si N és primer llavors exactament la meitat dels elements en $\{1,...,N-1\}$ satisfan $a^{(N-1)/2} \bmod N = 1$ i per tant la probabilitat que després de k iteracions la seqüència `b[1]=···=b[k]=1` és exactament 2^{-k}.

D'altra banda, en el cas que N sigui compost és possible que no hi hagi cap valor de a que satisfaci $a^{(N-1)/2} \bmod N = N-1$; en aquest cas el resultat serà el correcte. Tanmateix és possible que $a^{(N-1)/2} \bmod N = N-1$, però en aquest cas la meitat dels elements en $\{1,...,N-1\}$ satisfan $a^{(N-1)/2} \bmod N \notin \{1, N-1\}$, raó per la qual la probabilitat d'error serà 2^{-k}.

Pel teorema 3.1.4.2 es pot dir que l'algorisme 3.1.4.1 és d'error doble tal com la definició 1.1.2.1 indica, però millora el test de Solovay Strassen perquè amb el mateix marge d'error, no requereix el càclul del símbol de Jacobi, tot i així s'hauran d'esperar els resultats experimentals abans de treure cap conclusió.

Teorema 3.1.3.3 La complexitat de l'algorisme de Lehmann és $O\left(k \log^{1+\log_2 3} N\right)$

Prova

Si s'assumeixen 10 iteracions, l'únic que realment té un cost rellevant és l'exponenciació modular que pot ser duta a terme en $O(\log N)$ operacions aritmètiques; assumint que s'utilitza Karatsuba per multiplicar i dividir (la divisió és un cas particular de multiplicació en reals) es pot veure que:

$$O(\log N) \cdot O\left(\log^{\log_2 3} N\right) = O\left(\log^{1+\log_2 3} N\right)$$

3.1.4.4 Resultats experimentals

Tot seguit es presenten els resultats experimentals del test de Lehmann, després de 10 iteracions la probabilitat d'error queda reduïda a $9.765 \cdot 10^{-4}$. Intuitivament hauria de ser més eficient que el test de Solovay-Strassen però caldrà contrastar aquesta idea amb les següents taules confeccionades després de hores i hores d'execució.

Digits	Constant	Temps
25	2.60508	0:0:0:01
50	2.6325	0:0:0:07
60	2.60922	0:0:0:10
70	2.59715	0:0:0:14
75	2.7028	0:0:0:30
80	2.53503	0:0:0:14
85	2.52002	0:0:0:15
91	2.54884	0:0:0:21

Taula 3.22

Dígits	Constant	Temps
95	2.503	0:0:0:18
100	2.48088	0:0:0:18
127	2.46729	0:0:0:30
155	2.43464	0:0:0:40
175	2.42639	0:0:0:51
191	2.39349	0:0:0:51
200	2.42526	0:0:0:70
225	2.4375	0:0:1:01
253	2.41039	0:0:1:12
271	2.42898	0:0:1:50
289	2.42187	0:0:1:67
300	2.43196	0:0:1:96
355	2.43092	0:0:2:93
401	2.43645	0:0:4:10

Taula 3.23

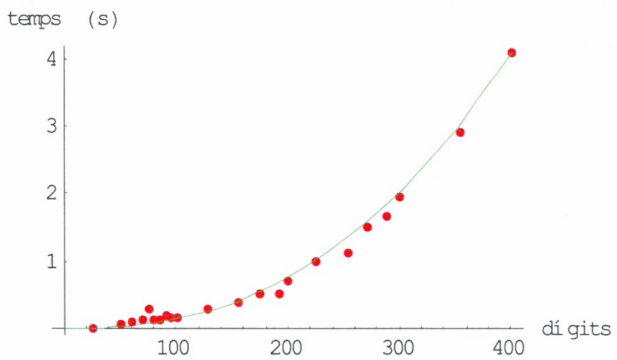


Figura 3.24

En aquesta gràfica es combinen els resultats experimentals amb el comportament teòric de l'algorisme, per tant no s'hauria d'allunyar de la funció:

$$f(x) = (x \log_2 10)^{2.5} / 10^8$$

El comportament dels valors de la constant és el següent:

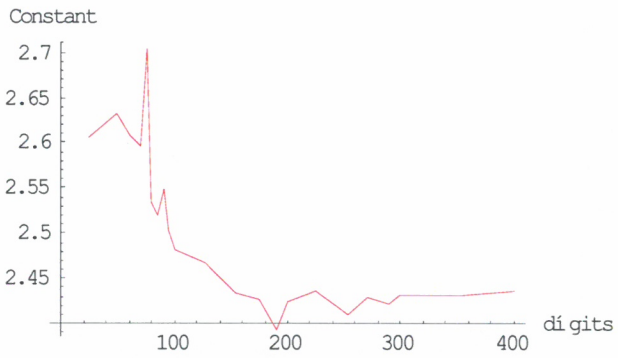


Figura 3.25

i com es veu es va estabilitzant.

Dígits	Constant	Temps
507	2.45201	0:0:8:15
601	2.44448	0:0:11:68
664	2.47528	0:0:18:89
777	2.45865	0:0:24:46
1000	2.4744	0:0:51:68
1137	2.47277	0:1:34:09
1276	2.48679	0:1:44:76
1281	2.51879	0:2:18:21
1505	2.50828	0:3:9:65
1781	2.52153	0:5:24:59
2000	2.5375	0:8:20:48
2506	2.52289	0:12:57:42
2993	2.55529	0:27:19:67
3235	2.56283	0:35:45:15
5000	2.59727	2:32:30:10
5941	2.57235	3:6:31:71
6002	2.5873	3:42:1:95

Taula 3.26

Els temps d'execució quan seguim augmentant la longitud de N segueix el següent patró:

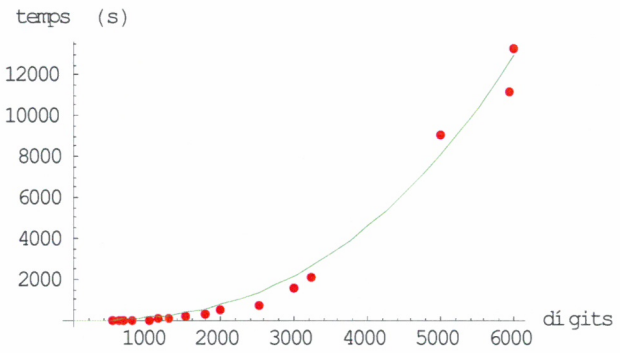


Figura 3.27

Cada cop s'assembla més a la funció $f(x) = (x \log_2 10)^{1+\log_2 3} / 10^7$, i la constant va convergint a $1 + \log_2 3$.

3.2 Test determinístic AKS

El 6 d'agost de 2002, Manindra Agrawal, Neeraj Kayal i Nitin Saxena van publicar a la pàgina web de l'Institut Indi de Tecnologia a Kapur el primer algorisme determinístic, incondicional i de temps polinòmic per verificar si un enter N és primer o no.

El títol, "PRIMES is in P", formula amb el llenguatge de la teoria de la complexitat computacional el que desenvolupa després durant l'article. PRIMES denota el problema de la primeritat i s'anomena "llenguatge" (que juntament amb les "funcions" és l'objecte d'estudi de la teoria de la complexitat). Per establir que un llenguatge pertany a P cal trobar un algorisme que per una entrada N realitzi $c(\log^k N)$ operacions i sigui determinístic.

Per tant, una persona amb coneixements d'aquesta ciència només llegir el títol de l'article ja sabia que s'havia trobat un algorisme polinòmic i determinístic per al problema mil·lenari de la primeritat. PRIMERS és un problema co-**NP** perquè si N no és primer té un certificat molt senzill, un factor no trivial o algun testimoni dels que s'han presentat en cada mètode pseudoaleatori. La pregunta de si PRIMERS pertany a P havia existit des del naixement de la terminologia i de la màquina Turing als anys 60.

Es farà un breu repàs als progressos fets per decidir si N és primer o no abans de veure les aplicacions pràctiques del algorisme. El primer mètode conegut data del 240 a.C, s'anomena garbell d'Eratosthenes i es basa en el fet que si N no té divisors en $[2..\sqrt{N}]$ és primer; té una complexitat de $O(\sqrt{N})$ i és per tant molt ineficient. El teorema petit de Fermat presenta un mètode força eficient de verificar si N és primer: donat N i un enter a tal que mcd(a,N)=1, $a^{N-1} \equiv 1 (\mathrm{mod}\, N)$ si N és primer. Hi ha però un tipus especial de nombres, com ja s'ha observat, que compleixen aquesta igualtat tot i no ser primers: són els nombres de Carmichael que reben el nom en honor al seu descobridor, el matemàtic nord-americà Robert Carmichael.

El 1976, Miller va proposar un test de primeritat determinista assumint la Hipòtesi de Riemann Extesa. Rabin va modificar el test de Miller poc després, convertint-lo en un test incondicional -en el sentit que no depenia de cap hipòtesi- però aleatoritzat i de temps polinòmic. Al mateix temps Solovay, amb la

col·laboració de Strassen, va obtenir un altre algorisme aleatoritzat polinòmic però basat en la llei de reciprocitat quadràtica.

El 1983 es va produir un avanç enorme; Adleman, Pomerance i Rumely van dissenyar el primer test determinístic incondicional però de temps superpolinòmic. És aplicable a N sense necessitat de conèixer la factorització de $N+1$ o $N-1$. El temps d'execució és calculable per unes constants $0 < C < C'$,

$$(\log N)^{C \log\log\log N} \leq t(N) \leq (\log N)^{C' \log\log\log N}$$

la seva correcció i anàlisi de complexitat està demostrat rigorosament a través d'eines de la teoria dels nombres com les lleis de reciprocitat; en part estava basat en la idea de Miller. Més tard, va ser redissenyat per Cohen i Lenstra per fer-lo més flexible. El 1986 Goldwasser i Kilian van proposar un test aleatoritzat que utilitzava corbes el·líptiques, la complexitat era polinòmica per la majoria de casos i produïa un certificat de primeritat, fins llavors tots els certificats eren de compositivitat.

Fins al 2002 havia estat impossible d'obtenir un algorisme incondicional, determinístic i polinòmic; l'aparició de l'AKS respon a "is PRIMES in P"? A nivell pràctic AKS no té cap repercussió significativa ja que ni pot ser utilitzat per factoritzar ni millora les prestacions dels algorismes aleatoritzats, que resulten molt eficients amb un error ínfim (però real). El nombre d'operacions aritmètiques a través de mètodes elementals de l'AKS és $O(\log^{14,5} N)$ i està basat en una generalització del teorema petit de Fermat, tot i que darreres optimitzacions hagin reduït el nombre d'operacions binàries a $O(\log^{7,5} N)$.

3 Conclusions

S'observa que es poden generar nombres primers d'una longitud de pocs milers de dígits decimals sense massa problemes utilitzant els algorismes aleatoritzats entre els què destaca especialment el de Miller-Rabin. Si n'haguessim de triar un, de ben segur que seria aquest: d'una banda per la seva eficiència i per l'altra el baix cost que té reduir l'error.

La primeritat de qualsevol nombre pot ser verificada en temps polinòmic en relació a la seva longitud amb un marge d'error molt baix. L'aplicació de la generació de primers és l'ús que d'aquests se'n fa en abundants criptosistemes però de fet el problema en sí ja té una gran bellesa. A principis d'aquests segle XXI, amb l'article de *Primes is in P* s'ha solucionat una pregunta històrica. S'ha de ser conscient del moment d'expansió tecnològica que s'esta produint tot i que no sempre tingui una aplicació pràctica directa.

A nivell experimental el resultat més sorprenent que s'ha obtingut ha estat el comportament assimptòtic del test de Solovay-Strassen que podia competir amb Miller-Rabin i Lehmann, superant-los en alguns casos. A primera vista es podia pensar la necessitat d'avaluar el símbol de Jacobi descartaria el test, tanmateix la llei de reciprocitat quadràtica provada per primer cop per Gauss s'ha revelat com un element definitiu a l'hora de dissenyar un algorisme que sigui capaç de calcular el símbol de Jacobi de forma eficient. Tanmateix, com ja s'ha senyalat, el marge d'error de Miller-Rabin li dona una posició preeminent que cap altre mètode aleatoritzat o determinístic està en condicions de qüestionar.

4. Factorització d'enters

El problema de distingir nombres primers dels compostos i de resoldre aquests darrers en els seus factors és considerat com un dels objectius més importants i útils en aritmètica... La dignitat de la ciència en ella mateixa sembla requerir que cada possible mitjà sigui explorat per obtenir la solució del problema tant elegant i tant famós.

Disquistiones Arithmeticae, article 329 (1801) Carl Friedrich Gauss

En el Teorema Fonamental de l'Aritmètica apareix també el problema cabdal de l'aritmètica: donat un enter N trobar la seva descomposició en factors primers. Els matemàtics com Euclides i Tales ja coneixien el problema, i encara avui l'home ha estat incapaç de trobar un mètode eficient per atacar el problema si bé és apassionant estudiar l'evolució dels apropaments a tant honorat problema.

Per aquesta raó molts criptosistemes moderns basen la seva seguretat en la intractabilitat del problema de la factorització; quan N és prou gran, trobar els factors primers requereix una gran quantitat d'operacions que es tradueix en llargs temps d'execució.

Fins ara l'home ha estat incapaç de dissenyar algorismes amb temps d'execució millors que $N^{o(1)}$; per tant, cal conformar-se amb complexitats subexponencials. L'objectiu d'aquesta secció és estudiar a nivell teòric els algorismes de Fermat, Lehmann, Pollard i Pomerance -aquest subexponencial- i tot seguit contrastar els resultats experimentals amb diversos nombres per decidir quin és el més apte per a cada cas.

4.1 Mètodes de complexitat exponencial

Aquest apartat està dedicat als algorismes de factorització amb complexitat exponencial, tenint en compte que n'hi ha de millors en principi no tindria massa sentit però existeixen raons per incloure aquesta part:

- Resulten eficients per nombres petits i són subrutines d'algorismes més eficients.

- El seu estudi i anàlisi han permès el desenvolupament dels algorismes de complexitat subexponecial que van començar a aparèixer la dècada dels 70.

- Mera curiositat de conèixer.

El mètode més conegut possiblement sigui el d'intent de dividir, pels enters en el rang $\left[3, \sqrt{N}\right]$; tanmateix quan N té una longitud de més de 40 dígits, realitzar un bucle d'aquest rang està fora de la capacitat computacional dels ordinadors actuals. Si N té factors petits, aquest mètode es presenta més eficient que alguns altres més elaborats, de manera que, de vegades, s'inclou dins d'altres algorismes.

4.1.1 Assaig de divisió

Al confrontar el problema de buscar divisors d'un nombre es pot tenir algunes consideracions per evitar càlculs innecessaris: tots parteixen de la teoria dels nombres i es basen en les propietats de certs primers. Per saber si un nombre és divisible per 2 només cal veure si el darrer dígit és parell; si la suma dels dígits de N és divisible per 3, aquest també serà un divisor de N; si l'últim dígit és zero i cinc, N serà divisible per 5; si la suma alterna dels dígits de N és congruent amb N mòdul 11, aquest serà un divisor... Com es pot observar els tests de divisibilitat esdevenen més complicats a mesura que l'ordre multiplicatiu del possible divisor augmenta.

Definició 4.1.1.1 L'ordre multiplicatiu és el valor mínim de k tal que

$$a^k \equiv 1 \pmod{N}.$$

Tanmateix el problema fonamental de l'aritmètica no admet dreceres, la solució en serà una d'elegant; i es pot afirmar perquè computacionalment aquestes consideracions de caràcter quasi anecdòtic són inútils, no produeixen cap efecte remarcable, fins i tot resulta més eficient provar la divisibilitat directament buscant el quocient i el residu.

Definició 4.1.1.2 El mètode d'anar provant divisors seqüencialment fins aconseguir un primer p que divideixi a N, a no ser que N sigui primer rep el nom d'intent de divisió.

Si els divisors que s'assagen no són primers sinó compostos, dels quals ja se n'ha provat la divisibilitat a través de la seva descomposició en factors primers, llavors es perd temps de manera inútil. Per accelerar el procés es poden tenir en compte alguns aspectes que faciliten la tasca -a nosaltres o bé a l'ordinador-: tots els primers són imparells excepte el dos. Només per aquest fet es pot eliminar el 50% dels nombres; si no es troba cap divisor arribats a $\sqrt{N'}$, llavors es pot aturar-nos perquè N' serà primer. Si es disposa d'una llista de primers fins a $\sqrt{N}$ es pot veure pel Teorema dels Nombres Primers que el nombre d'operacions aritmètiques serà $O\left(2\sqrt{N}/\ln N\right)$.

L'ús directe del mètode per intentar assolir factoritzacions completes de N està força limitat a causa de la capacitat computacional dels ordinadors actuals; com a molt es pot factoritzar nombres de 20 dígits en un dia sencer considerant 10^6 divisions per segon. Però aquest mètode és molt útil en subrutines per obtenir factoritzacions parcials o saber si un nombre és B-uniforme; és a dir, si té divisors menors que B.

```
Algorisme 3.1.3
ENTRADA: un enter N
SORTIDA: el conjunt D de divisors
BEGIN
01 D:={''};
02 n:=N; d:=2;
03 While (n mod d=0) do
04  begin
06   D:=D+{2}; n:=n div 2;
07  end; d:=3;
08 while (d^2<n) do
09  begin
10   while (n mod d=0) do
11    begin
12     n:=n div d; D:=D+{d};
13    end;
14   d:=d+2;
15  end;
END.
```

Definició 4.1.1.3 Si N un enter tal que el seu segon major factor primer no superi N^c.

Crandall i Pomerance en [9], basant-se en la publicació de Knuth i Pardo [27], observen que el valor de c tal que en un conjunt la densitat assimptòtica d'aquests nombres sigui de $1/2$ és la solució a l'equació:

$$\int_c^{1/2} \frac{\ln(1-u) - \ln u}{u} du = \frac{1}{2}$$

Per mètodes numèrics s'ha pogut aproximar el valor de c sense donar una expressió tancada

$$c \approx 0.23043660131599974571471085700604655750807 54...$$

Mathematica 5.2 dona un error del tipus `General::ivar:` i no permet obtenir més decimals. Aquestes observacions de Crandall, Pomerance, Knuth i Pardo són extremadament interessants perquè posen de relleu el fet que els nombres compostos amb factors primers grans són rars.

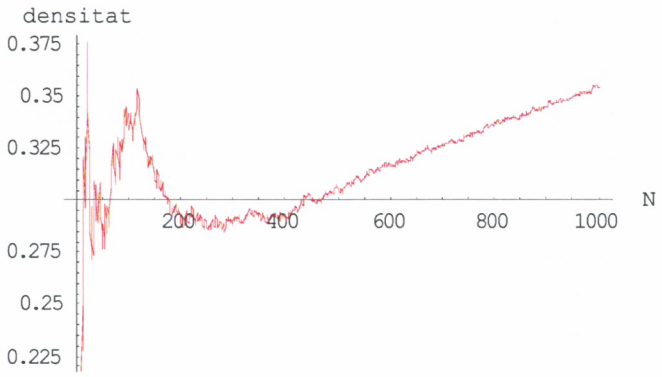


Figura 4.1

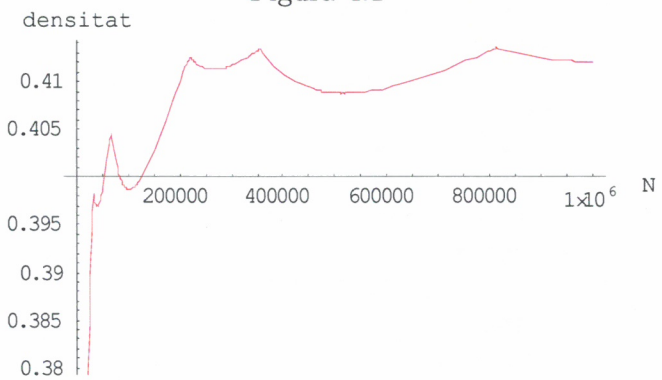


Figura 4.2

A mesura que la N augmenta, la densitat de nombres que compleixen la definició 4.1.1.3 va convergint a una meitat.

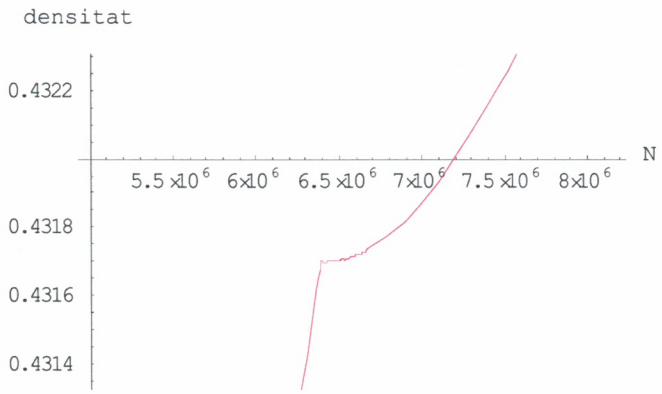


Figura 4.3

A títol informatiu, i per donar una idea del volum de calcul necessari per confeccionar aquestes taules, establim que la Figura 4.1 ha requerit 43.687 segons en Mathematica 5.2, la Fig. 4.2 11 623 segons i la Fig. 4.3 16 078.7 segons.

4.1.2 Mètode de Fermat

L'objectiu d'aquest algorisme de factorització és aconseguir expressar N com:

$$N = a^2 - b^2$$

per $0 < (a,b) \in \mathbb{Z}$, de manera que

$$N = (a+b)(a-b) = xy$$

si $(a-b)$ ¿1 la factorització no és trivial. Si $N = xy$ es pot veure fàcilment que $a = \frac{1}{2}(x+y)$ i $b = \frac{1}{2}|x-y|$ tenint en compte que $x = (a+b); \quad y = (a-b);$ per tant:

$$\tfrac{1}{2}(x+y) = \tfrac{1}{2}\big((a+b)+(a-b)\big) = a$$
$$\tfrac{1}{2}(x-y) = \tfrac{1}{2}\big((a+b)-(a-b)\big) = b$$

S'haurà de buscar un quadrat perfecte solució a l'equació

$$b = \sqrt{a^2 - N};$$

s'ha de començar amb valors de a $\lceil \sqrt{N} \rceil$, per evitar nombres negatius, i a partir d'aquí s'ha d'anar incrementant 1 fins a trobar un quadrat perfecte. La seqüència és finita en el sentit que de ben segur es trobarà una solució abans de $a = (N+1)/2$ perquè:

$$a^2 - N = b^2$$
$$\big((N-1)/2\big)^2 = \big((N+1)/2\big)^2 - N$$
$$\big((N+1)/2\big)^2 - \big((N-1)/2\big)^2 = N$$

En aquest cas només s'obté la factorització trivial de N quan $N = 1 \cdot N$. Es pot limitar doncs el valor de a fins $a = (N+9)/6$, perquè si N és senar no pot tenir 2 com a factor. Si a arriba fins $(N+9)/6$ i no apareix cap quadrat perfecte llavors N queda provat que és primer.

Quan els factors són petits aquest mètode és pitjor que els intents de dividir ja que mentre que aquell té complexitat de $O\left(\sqrt{N}\right)$ operacions aritmètiques , el mètode de Fermat en té $O(N)$ pel pitjor cas, per tant sovint es combinen per donar millors resultats. Aquesta complexitat és el resultat d'avaluar pel pitjor cas el límit de

$$\lim_{N\to\infty} \frac{N+9}{6} - \sqrt{N} = N$$

Es diu que és un algorisme de complexitat exponencial perquè és lineal en N i exponencial en la mida que s'expressa com b^N .

Algorisme 4.1.2.1 de Fermat
INPUT: N **enter** >1
OUTPUT: X,Y **enters** ≥ 1

BEGIN
For a:=$\left\lceil \sqrt{N} \right\rceil$ **to** $(N+9)/6$ **do**
If $\left(b = \sqrt{a^2 - N} \in \mathbb{Z}\right)$ **then** return:=(a-b,' ', a+b)
Return:=('1 i N, és primer');
END.

Es poden aplicar altres consideracions de caire aritmètic: Si s'intenta factoritzar un nombre tal que $N = 11 \cdot 47 = 517$, es pren el primer valor de a en $a = 23$. Aquest algorisme passa per alt el fet que $N \equiv 1 (\mathrm{mod}\, 4)$ i $N \equiv 1 (\mathrm{mod}\, 3)$, per tant descuida que a no pot ser múltiple de 3 ni de 2 com efectivament es pot comprovar ja que a=29 i b=18.

Es pot utilitzar congruències que permetin estalviar el càlcul d'arrels, per exemple: si $N \equiv 1 (\mathrm{mod}\, 4)$ llavors a ha de ser senar o bé si $N \equiv 2 (\mathrm{mod}\, 3)$ llavors a ha de ser múltiple de 3...

L'algorisme de Fermat funciona millor quan els dos factors són d'un valor similar i es troben propers a $\sqrt{N}$. Per tant es pot d'alguna manera incidir en N per tal de que compleixi aquesta condició i per reduir el temps d'execució.

Hi ha dues maneres de dur a terme aquesta idea: si es té $N = ab$ es multiplica per una constant tal que $N' = kN = a'b'$ -cal veure que a, b, a', b' poden no ser primers-, ara es calcula el $\mathrm{mcd}(a', N) = a$ i el $\mathrm{mcd}(b', N) = b$, s'obté doncs els dos factors de N.

L'altra mètode suposa que es coneix la factorització i intenta establir una relació entre els dos factors en termes d'un racional com $\frac{a}{b}$; a més pren dos escalars com c i d establint també una relació $\frac{c}{d}$. Ara cal veure que si

$$\frac{c}{d}N = ab\frac{c}{d}$$
$$Ncd = abcd$$

llavors $\mathrm{mcd}(bc, N) = b$ i $\mathrm{mcd}(ad, N) = a$. Aquest sistema funciona mentre b no divideixi a d ni a divideixi a c. Tot i que aparentment aquestes consideracions semblin irrellevants ja que l'objectiu de l'algorisme és trobar la factorització no trivial - si existeix-, són la base d'un altra algorisme proposat per Lehmann i que d'alguna manera formalitza totes aquestes idees més o menys inconexes que han anat sorgint.

4.1.3 Mètode de Lehman

En aquest algorisme es formalitza l'elecció d'un multiplicant per accelerar el mètode de Fermat. De fet el combina amb l'assaig de divisió i aprofita l'informació que donen els residus.

El funcionament de l'algorisme es recolza profundament en la teoria dels nombres. Veure el pseudocodi sense entendre perquè funciona i quines són les bases matemàtiques seria tenir-ne una visió incompleta. Per tant, a partir de l'article publicat per R. Sherman Lehmann en la revista *Mathematics of Computation* l'abril de 1973, es precedeix a analitzar la complexitat i la correcció.

Teorema 4.1.3.1 Per qualsevol $\xi \in \mathbb{R}$ i N un enter positiu, existeix una fracció h/k irreductible tal que si $0 < k \leq N$

$$\left|\xi - \frac{h}{k}\right| \leq \frac{1}{k(N+1)},$$

o el que és el mateix, per qualsevol màxim B, existeixen dos enters u i v tal que $v \leq B$ i $\left|\frac{u}{v} - \frac{p}{q}\right| < \frac{1}{vB}$.

Prova
Es remet el lector a *Theory of Numbers* G. H. Hardy i E. M. Wright. La demostració es fa a través de sèries de Farey i és el teorema 36, (veure [25]).

Algorisme 4.1.3.2 de Lehmann
INPUT: N **enter** >21
OUTPUT: X,Y **enters** ≥ 1

BEGIN
for d:=1 **to** $\lceil N^{1/3} \rceil$ **do**
if (d mod N)=0 **then** return d;

for k:=1 **to** $\lceil N^{1/3} \rceil$ **do**
for a:= $\lceil 2\sqrt{kN} \rceil$ **to** $\lceil 2\sqrt{kN} \rceil + N^{1/6} / \left(4\sqrt{k}\right)$ **do**
if $\left(b = \sqrt{a^2 - 4kN}\right) \in \mathbb{Z}$ **then return mcd**(a+b,N);

END.

Teorema 4.1.2.3 L'algorisme 4.1.3.2 funciona per $N \geq 21$.

Prova
Assumim que N no és divisible per cap enter inferior a $N^{1/3}$, llavors si és compost tindrà dos factors tal que $N^{1/3} < p \leq q$. S'utilitza una variable $k \leq \lceil N^{1/3} \rceil$ amb dos factors primers, v i u.

Recuperant el teorema 36 i multiplicant tots dos costats de la igualtat per v i per q s'obté la següent igualtat si $B = N^{1/6}\sqrt{q/p}$:

$$|uq - vp| < \frac{q}{N^{1/6}\sqrt{q/p}}$$

es pot demostrar que el costat dret de la desigualtat és igual a $N^{1/3}$

$$\frac{q}{N^{1/6}\sqrt{q/p}} = N^{1/3}$$
$$\frac{\sqrt{pq}}{\sqrt{q}} = N^{1/2}$$
$$N^{1/2} = N^{1/2}$$

de manera que s'obté finalment,

$$|uq - vp| < N^{1/3}$$

Però s'ha de demostrar que $k = uv \leq \lceil N^{1/3} \rceil$. Altra vegada s'utilitza el teorema 36 una mica transformat,

$$\left|\frac{u}{v} - \frac{p}{q}\right| < \frac{1}{vB}$$
$$\frac{u}{v} < \frac{p}{q} + \frac{1}{vB}$$

per tota $v \leq B$. D'aquesta manera es pot es pot observar que:

$$k = uv = \frac{u}{v}v^2 < \frac{p}{q}v^2 + \frac{v}{B} \leq \frac{p}{q}\frac{q}{p}N^{1/3} + 1 = N^{1/3} + 1$$

Simplement cal veure que en el primer pas es multiplica per v^2 tots dos costats de la igualtat. Després es suposa $v = B = N^{1/6}\sqrt{q/p}$ perquè aquest és el pitjor cas, si es compleix, la resta també de manera que $k = uv \leq \lceil N^{1/3} \rceil$.

S'han de definir dues altres variables, a i b tal que: $a = uq + vp$ i $b = |uq - vp|$. Es pot rescriure $4kN$ de la següent manera:

$$\begin{aligned} 4kN = 4uvpq &= (uq + vp)^2 - (uq - vp)^2 \\ &= \left(u^2q^2 + 2vpuq + v^2p^2\right) - \left(u^2q^2 - 2vpuq + v^2p^2\right) = a^2 - b^2 \end{aligned}$$

Aquest fet permet intuir que altra vegada es podrà aplicar el mètode de Fermat per factoritzar a través dels quadrats perfectes. Si $4kN = a^2 - b^2$ queda veure el rang de a en el què s'ha de treballar per tal que $\sqrt{a^2 - 4kN} \in \mathbb{Z}$.

Com que:

$$4kN = a^2 - b^2$$
$$a^2 = 4kN + b^2$$
$$a^2 \geq 4kN$$

és fàcil de veure que $a = uq + vp \geq 2\sqrt{kN}$. Per tant $a = 2\sqrt{kN} + C$. La qüestió doncs és quin serà el màxim valor de C. De l'expressió

$$4kN + 4C\sqrt{kN} \leq \left(2\sqrt{kN} + C\right)^2 = a^2 = 4kN + b^2 < 4kN + N^{2/3}$$

en deduïm que: $4C\sqrt{kN} < N^{2/3}$. L'última desigualtat es basa en el fet que es pot provar que $b < N^{1/3}$: cal recordar que $b = |uq - vp|$ i que anteriorment assumint el teorema 36 de Hardy i Wright,

$$|uq - vp| < N^{1/3}.$$

I encara se'n pot extreure una altra conclusió d'aquesta expressió, es pot determinar el valor de C! De manera que en l'algorisme es pugui limitar de manera més precisa els possibles valors de a, en el rang $2\sqrt{kN} \leq a < 2\sqrt{kN} + C$.

Es calcula C si $4C\sqrt{kN} < N^{2/3}$:

$$C < \frac{N^{2/3}}{4\sqrt{kN}} = \frac{N^{4/6}}{4\sqrt{k}N^{3/6}} = \frac{N^{1/6}}{4\sqrt{k}}$$

Si es detecta una valor de a per al qual l'arrel produeix un enter, s'ha de demostrar que quan es calcula $\text{mcd}(a+b, N)$ és un factor no trivial de N. Cal recordar que N divideix a $4kN = a^2 - b^2$, per tant dividirà a $(a+b)(a-b)$ i per demostrar que s'ha trobat un factor trivial és suficient de veure que $a + b < N$. Es suposa el pitjor cas de manera que la resta cauran pel seu propi pes.

El valor de b més gran es produeix quan $a = \left\lfloor 2\sqrt{kN} + N^{1/6} / \left(4\sqrt{k}\right) \right\rfloor$ i $k = \left\lceil N^{1/3} \right\rceil$.

Si s'evalua $a^2 - 4kN$ per aquests paràmetres s'obté quelcom:

$$\left(2\sqrt{kN}+\frac{N^{1/6}}{4\sqrt{k}}\right)^2-4kN=\left(4kN+\frac{4\sqrt{kN}N^{1/6}}{4\sqrt{k}}+\frac{N^{1/3}}{16k}\right)-4kN$$
$$=N^{1/2}N^{1/6}+\frac{N^{1/3}}{16k}=N^{4/6}+\frac{N^{1/3}}{16\left(N^{1/3}+1\right)}\simeq N^{2/3}$$

al prendre l'arrel se sap del cert que $b<N^{1/3}$. Com es coneix el valor màxim de a s'escriu la següent expressió:

$$a+b<2\sqrt{kN}+\frac{N^{1/6}}{4\sqrt{K}}+N^{1/3}<2\sqrt{\left(N^{1/3}+1\right)N}+\frac{N^{1/6}}{4\sqrt{\left(N^{1/3}+1\right)}}+N^{1/3}<N$$

Per veure la darrera desigualtat es veu que es compleix per $N\geq 21$

$$\lim_{N\to\infty}2\sqrt{\left(N^{1/3}+1\right)N}+\frac{N^{1/6}}{4\sqrt{\left(N^{1/3}+1\right)}}+N^{1/3}=\infty$$

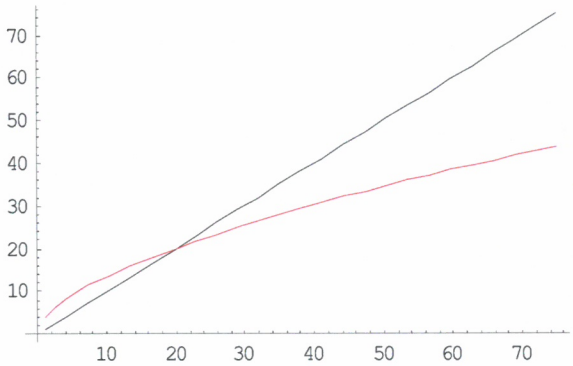


Figura 4.4
Comparativa entre la funció N i la desigualtat.

□

Teorema 4.1.2.4 L'algorisme té una complexitat de $O\left(N^{1/3}\ln\ln N\right)$

Prova

L'anàlisi de la complexitat del primer *for* és molt senzill, $O\left(N^{1/3}\right)$. El segon ja requereix més precisió. Es pot estimar ràpidament si l'arrel és un enter a través d'un algorisme amb un cost de $O(\ln \ln N)$, que es comentarà tot seguit. Recordar també que els quadrats perfectes tenen unes característiques que poden ajudar tot i que en aquesta implementació bàsica no es tindrà en compte. El nombre d'iteracions del *for*, si no s'ha pogut factoritzar en la primera part, serà:

$$\lim_{N\to\infty} \sum_{k=1}^{\lceil N^{1/3} \rceil} \left(\frac{N^{1/6}}{4\sqrt{k}} + 1 \right) = N^{1/3} = O\left(N^{1/3}\right)$$

Aquest, en realitat, és el nombre de vegades que es crida l'algorisme per calcular la part entera d'una arrel, per tant la complexitat total és $O\left(N^{1/3} \ln \ln N\right)$ si no ha pogut ser factoritzat en l'intent de divisió. En l'anàlisi assumim que l'únic que té un cost és l'algorisme que calcula si l'arrel és un enter i l'intent de dividir, però al fer el límit, desapareix.

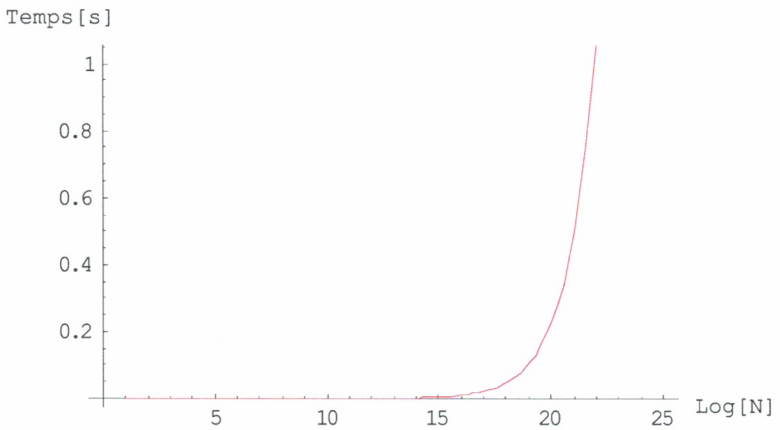


Figura 4.5

L'eix de les abscisses és logarítmic. Apareix doncs la potència de 10 de N mentre que les ordenades representen el temps(s) tenint en compte que es poden realitzar 10^8 operacions binàries per segon.

Sembla prou eficient per enters menors a 10^{25}, però cal ser conscients que per enters grans és inviable:

$N=10^x$	Temps
25	10 s
30	8 minuts
40	2 setmanes
50	85 anys
75	$1.95 \cdot 10^{10}$ anys
100	$4.4 \cdot 10^{18}$ anys

Taula 4.6

No crec que cap de nosaltres veiés N factoritzat a través d'aquest mètode quan x és major de 50 dígits. La clau de l'algorisme resideix en una funció auxiliar per calcular la part entera de N que és molt eficient. La variació dels seus valors té el següent comportament si s'assumeix: $f(x) = \ln \ln 10^x / 10^8$.

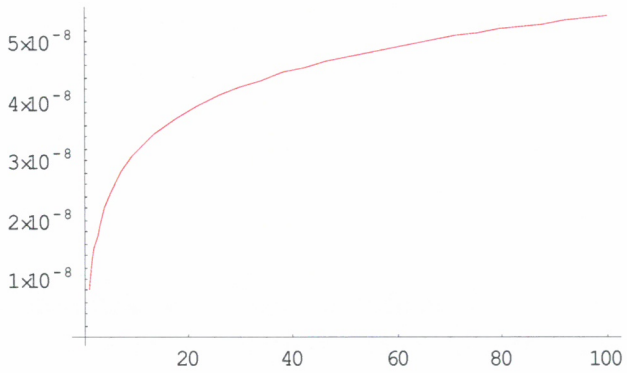


Figura 4.7

Per estudiar aquest algorisme s'ha partit d'un pressupòsit inicial, el teorema de Hardy i Wright. Tanmateix el text no ha perdut valor per aquest fet ja que el treball es centra, en la mesura del possible, en l'algorisme. A més molts articles d'investigació matemàtica, es basen en l'assumpció d'una hipòtesi com la de Riemann o d'altres teoremes que els permeten avançar en la seva investigació. Tot seguit es presenta aquest algorisme enormement eficient, que es pot executar per valors molt grans.

A través de diverses iteracions, amb el mètode de Newton:

$$x_{N+1} = x_N - \frac{f(x)}{f'(x)}$$

al desenvolupar per $f(x) = x^2 - a$ s'obté

$$x_{N+1} = \frac{x_N}{2} + \frac{a}{2x_N}$$

que permetrà calcular $\sqrt{a}$. Aquestes consideracions permeten plantejar l'algorisme següent.

```
Algorisme 4.1.3.5
INPUT: N enter
OUTPUT: X enter ≥1

BEGIN
 x := 2^⌊B(N)/2⌋;
 while (y < x) do
  begin
   y:=⌊(x + ⌊N / x⌋) / 2⌋;
   x:=y;
  end;
 return x;
END.
```

On B(N) representa el nombre de exacta de dígits de la representació binària, és millor que una simple aproximació a través del logarítme base 2. Després de ser executat, per saber si N és un quadrat perfecte es verifica si es compleix aquesta expressió: $N - x^2 = 0$.

Teorema 4.1.2.6 Es pot calcular la part entera de l'arrel en $O(\ln \ln N)$

4.1.4 Algorismes de Pollard

En la dècada dels 70 James Pollard va introduir els mètodes de Monte Carlo - de fet va ser ell qui els donà el nom-. Entre els diversos que va dissenyar destaquen especialment el mètode de *rho*, el *p*-1 i els basats en logaritmes discrets. L'interés es centra en el mètode de *rho* i el *p*-1 perquè són força coneguts i no requereixen coneixements de logaritmes discrets.

Teorema 4.1.3.1: En un conjunt finit de N elements, la probabilitat que al prendre dos elements aquests siguin iguals serà superior al 50% a partir de $\sqrt{N}$ intents.

Prova: Primer es calcula la probabilitat que aquests dos elements extrets no siguin iguals. A partir d'aquí se'n dedueix la probabilitat que sí siguin iguals perquè les afirmacions són complementàries. Per N elements i k intents:

$$P_1 = 1 \cdot \left(\frac{N-1}{N}\right) \cdot \left(\frac{N-2}{N}\right) \cdots \left(\frac{N+1-k}{N}\right)$$
$$= \frac{N!}{N^k (N-k)!}$$

Com s'ha dit anteriorment les afirmacions són complementàries de manera que si es pot calcular la probabilitat de que no siguin iguals ràpidament s'obté la probabilitat que sí siguin iguals de la següent manera: $P_2 = 1 - P_1$. Sorprenentment la probabilitat de que siguin iguals serà major al 50% a partir de $k = \sqrt{N}$ iteracions o intents.

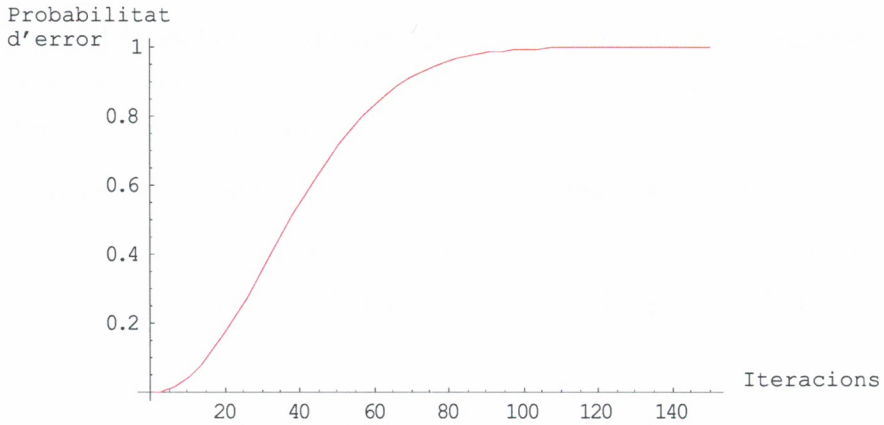


Figura 4.8

En aquest cas N=1000, i es veu que la propietat és superior a 50% a partir de $\sqrt{N} \simeq 31.622$.

4.1.3.1 Mètode Rho de Pollard

El mètode d'assaig de divisió en un nombre N suposat compost prendrà pel cap baix $p/\ln p$ divisions on p és el factor primer més petit. El mètode està basat en tres idees fonamentals: es pot considerar funcions amb congruències per

generar seqüències pseudoaleatòries, l'algorisme del màxim comú divisor és molt eficient i per últim la seqüència serà cíclica i tindrà un període aproximat de $\sqrt{N}$. Aquest fet es dedueix per la paradoxa de l'aniversari de la teoria combinatòria elemental que s'acaba d'explicar.

Si s'utilitza una funció F qualsevol que actuï tingui en un rang limitat sigui $\{0,1,...,N-1\}$, perquè és cíclica es pot anar iterant.

$$F_{r+1} = F(F_r) (\mathrm{mod}\, N)$$

S'espera que comenci a repetir valors abans de $O\left(\sqrt{N}\right)$ iteracions, és a dir, per $0 \le j < k = O\left(\sqrt{N}\right)$, $F_{(j)}(x) = F_{(k)}(x)$. El problema principal és que no es coneix p; és el valor que s'està buscant de manera que s'ha de jugar amb el fet que: $a \bmod p \equiv a \bmod N \bmod p$ quan p és el major factor primer de N. Se sap doncs que quan $F_r(x) \equiv F_k(x) \bmod p$ llavors es complirà que $\mathrm{mcd}\left(F_r(x) - F_k(x), N\right) = p$ o q.

Obtinguda aquesta relació, en l'algorisme prova valors de la funció fins que aparegui un factor no trivial. La probabilitat que es produeixin dos successos seguits és el que s'anomena probabilitat composta, la probabilitat que es produeixi el primer multiplicat pel segon: en aquest cas, $\left(\sqrt{N}\right)^2 = N$. Si haguéssim d'explorar totes les parelles les $N/2$ parelles, i executant cada cop com a subrutina l'algorisme d'Euclides aquest mètode seria molt pitjor que l'assaig de divisió.

Aquí és on entra el mètode per trobar cicles atribuït a Floyd: sigui $L = k - j$ i qualsevol $m \ge j$ llavors

$$F_{(m)}(x) \equiv F_{(m+l)}(x) \equiv F_{(m+2l)}(x) \equiv F_{(m+3l)}(x) \equiv ... \equiv F_{(m+kl)} (\mathrm{mod}\, p)$$

quan m pren el valor del primer múltiple de l superior a j, llavors

$$F_{(m)}(x) \equiv F_{(2m)}(x) (\mathrm{mod}\, p) \text{ i } m \le k = O\left(\sqrt{p}\right).$$

S'obtindrà la factorització abans de $O\left(\sqrt{p}\right)$ passos; per tant l'algorisme és exponencial en la longitud de N. Només pot ser utilitzat quan p és petit de

manera que és impracticable per trencar els criptosistemes actuals. Curiosament va servir per factoritzar el nombre $F_8 = 2^{2^8} + 1 = 2^{256} + 1$ perquè tenia un factor primer petit.

Es pot dur a terme una important discussió sobre quina és la millor expressió per a la funció F i això permet fer una mica d'experimentació a través l'implementació de l'algorisme següent.

Mètode Rho de Pollard

```
INPUT: enter N compost
OUTPUT: factor primer

BEGIN
1. A:=random [1..N-3];
   S:=random [1..N-1];
   U:=V:=S;
   F(x):=(x*x+a) mod N;

2. U:=F(U);
   V:=F(V);
   P:=mcd(U-V,N);
   if (P=1)goto 2;

3. if (P=N)goto 1;

4. retorna p;      {factor no trivial}
END.
```

Succeeix sovint que la teoria divergeix de la pràctica: evidentment executar un algorisme probabilístic requereix una font d'atzar, però els ordinadors són unes màquines elevadament determinísitques per tant obtenir atzar és una quimera. Per tant cal conformar-se amb seqüències pseudoaleatòries obtingudes a partir de generadors diferents.

Quan s'ha implementat l'algorisme s'ha utilitzat un generador linear de congruències determinat per

$$x_{N+1} = (ax_N + b) \bmod M$$

aquest generador té un període M si: $\text{mcd}(b, M) = 1$, $4 | a - 1$ si $4M$ i quan tots els factors primers de M divideixen a-1. x_N denota la llavor i en criptografia és fonamental que romangui secreta.

En aquest cas s'ha utilitzat $M = M_{127} = 2^{127} - 1 = 17014...84105727$, com que és primer, evito complicacions amb les altres propietats, i cap nombre que intenti factoritzar amb aquest mètode tindrà 39 dígits per raons de temps

d'execució. Simplement cal veure que b i M siguin coprimers. Es pren $b = 7987431212130510104$.

4.1.3.2 Mètode $p-1$

Aquest algorisme es basa en dues idees fonamentals: el teorema petit de Fermat que ja s'ha tractat i segons el qual

$$a^{p-1} \equiv 1 \pmod{p}$$

on p és un primer senar i el concepte de *B-uniforme*. L'objectiu aquí torna a ser factoritzar N i poder retornar un factor no trivial de N. Si p-1 és *B-uniforme,* si k és el producte de tots els nombres primers iguals o menors que B, llavors k és un múltiple de p-1 de manera que $a^k \equiv 1 \pmod{p}$, i si p és un factor primer de N llavors p divideix el $\text{mcd}\left(a^k - 1 \pmod{N}, N\right)$.

Aquestes consideracions teòriques no són suficients perquè quan s'executa l'algorisme es desconeix la factorització de p-1 i tampoc se sap quin ha de ser el valor de B. Per tant a l'inici s'escolleix un valor qualsevol de B que no sigui massa gran i s'espera tenir sort; avui en dia però, tenint en compte la magnitud de N, la probabilitat de factoritzar N és remota. Tot i així el descobriment de l'algorisme va inquietar els directius de RSA i es va recomanar que els factors primers de N tinguessin com a mínim un primer gran que factoritzés p-1. En la literatura que rodeja la criptologia es coneixen aquest tipus de primers com *primers forts.*

La complexitat de l'algorisme p-1 de Pollard és $O\left(p_r^e \middle| p-1\right)$, és a dir la major potència d'un primer que divideixi a p-1. L'esquema es pot reelaborar aplicant p+1.

Mètode p-1 de Pollard

INPUT: enter compost **N**
OUTPUT: factor trivial o primer

BEGIN
1 Trobar la seqüència de primers tal que
$p_1 < p_2 < \ldots < p_r \leq B$ i per cada primer
el valor de a tal que $p_i^{a_i} \leq B$

2 C:=random$\{2, \ldots, N-1\}$
for i:=1 to m **do**
for j:=1 to a_i **do** $c := c^{p_i} \bmod N$;

3 $d := mcd(c-1, N)$;
return d;
END.

Richard Brent, del laboratori de computació a la Universitat d'Oxford, ha donat una formula experimental que permet aproximar l'any A en el qual es podrà factoritzar un nombre de D dígits decimals:

$$A = 13,24D^{1/3} + 1928,6\,.$$

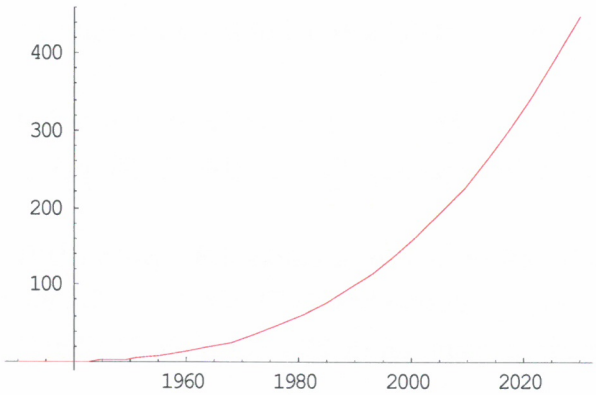


Figura 4.9

La formula es basa en la llei de Moore segons la qual el poder computacional es doble cada any i mig i no té en compte el fet que es pugui fer algun avanç revolucionari; considera la complexitat de l'algorisme *Number Field Sieve* que és una optimització de la *Quadratic Sieve*. La gràfica representa la previsió en un termini de 30 anys. Anar més enllà és inútil perquè de ben segur apareixeran nous algorismes.

Historia de la factorització

1588-1648 Marin Mersenne creu haver trobat un primer amb 2^{67}-1=147 573 952 589 676 412 927.

1842-1891 Edouard Lucas prova que 2^{67}-1 ha de ser el producte de dos nombres però no els pot calcular.

1903 Nelson Cole presenta la factorització de 2^{67}-1 els factors eren: 193 707 721 i 761 838 257 287 però va passar els diumenges de tres anys per calcular-los.

1970 Michael A.Morrison i John Brillhart aconsegueixen la factorització del setè nombre de Fermat, $F_7 = 2^{128} + 1 = p_{17}p_{22}$.

1975 Brillhart declarara a la revista Mathematics of Computation que la factorització d'un nombre de 50 dígits seria irrealitzable.

1983 Amb una xarxa d'ordinadors J.A. Davis i D.B. Holdridge factoritzen un nombre de 71 dígits per obtenir els dos factors primers, un de 30 i l'altre de 41 dígits.

1986 R.D.Silverman factoritza un nombre de 75 dígits.

1988 Silverman factoritza un nombre de 90 dígits M.S.Massane i Arjen Lenstra aconseguir la primera factorització d'un nombre de 100 dígits.

1989 A.Lenstra dissenya el seu mètode de corbes el·líptiques per factoritzar un nombre de 106 dígits.

1990 Pollard desenvolupa el *NFS* i Lenstra i Massane factoritzen un nombre de 155 dígits: $2^{512} + 1$.

1999 La factorització d'un nombre de 140 dígits a través de *NFS* requereix solucionar una matriu de 4 671 181 files i 4 704 451 columnes l'equivalent a 810 MB de memòria i 100 hores de CPU! (la dificultat resideix en la confecció de la matriu) .

2001 El QS factoritza un nombre de 135 dígits de la forma $2^{803} - 2^{402} + 1$. És dels últims grans nombres que factoritza perquè ja està sent substituït per *NFS* i *SNFS*.

2003 El mètode de Pollard p-1 és utilitzat per Phil Zimmermann per trobar factors de 57 dígits de dos nombres diferents: $6^{396} + 1$ i $11^{260} + 1$.

2005 Es factoritza el nombre RSA-640 de 200 dígits decimals.

4.5 Comparativa entre els diferents mètodes

Tot seguit es procedeix a executar una implementació per *Mathematica* 5.2 dels diversos mètodes de factorització. Un cop mostrats els resultats se n'extrauran conclusions i es decidirà quin mètode s'escau millor en cada cas.

p	q	N	Temps
4 133	6 121 607	25 300 601 731	0:00:11:11
2 741	67 282 409	184 421 083 069	0:02:04:17
3	10 093	30 279	0:00:00:15
3	84 189 997	252 569 991	0:02:34:07
4 133	6 121 607	25 300 601 731	0:00:11:31
681 647	948 547	646 574 216 909	0:00:01:78
77 137	454 759	35 078 744 983	0:00:00:84
512 849	670 211	343 717 041 139	0:00:01:25
136 273	489 571	66 715 308 883	0:00:00:91
210 643	640 727	134 964 657 461	0:00:01:19
22 165 237	26 263 157	582 129 099 273 209	0:00:48:55
300 961	25 507 091	7 676 639 614 451	0:00:46:81
16 518 833	33 374 347	551 305 264 577 051	0:00:01:45
1 236 583	19 471 313	24 077 894 643 479	0:00:35:57
10 710 431	27 074 389	289 978 375 251 659	0:00:50:11
1	6 389 351	6 389 351	0:00:11:73
1	562 577	562 577	0:00:01:03
1	1 859 441	1 859 441	0:00:03:42
33 573 653	62 022 901	2 082 335 356 227 353	0:01:54:41
277 443 083	548 425 091	152 156 748 041 595 553	0:16:42:81
276 173 267	419 566 859	115 873 150 174 958 353	0:13:00:69
757 817 069	840 664 129	637 069 626 252 217 901	0:25:50:16
296 254 423	470 032 573	139 249 228 705 320 379	0:14:24:70
700 232 219	997 014 107	698 141 400 518 913 433	0:30:31:61

Taula 4.10

Resultats obtinguts a través del mètode d'assaig de divisó.

p	q	N	Temps
4 133	6 121 607	25 300 601 731	0:08:40:73
2 741	67 282 409	184 421 083 069	1:30:33:27
3	10 093	30 279	0:00:00:15
4 133	6 121 607	25 300 601 731	0:11:17:51
681 647	948 547	646 574 216 909	0:00:01:48
77 137	454 759	35 078 744 983	0:00:11:08
512 849	670 211	343 717 041 139	0:00:00:73
136 273	489 571	66 715 308 883	0:00:07:73
210 643	640 727	134 964 657 461	0:00:08:14
22 165 237	26 263 157	582 129 099 273 209	0:00:12:31
300 961	25 507 091	7 676 639 614 451	0:24:49:39
16 518 833	33 374 347	551 305 264 577 051	0:03:27:10
1 236 583	19 471 313	24 077 894 643 479	0:13:04:73
10 710 431	27 074 389	289 978 375 251 659	0:04:25:71
1	6 389 351	6 389 351	0:02:38:64
1	562 577	562 577	0:00:13:27
1	1 859 441	1 859 441	0:00:58:03
1	742 609 451	742 609 451	0:32:04:28
$10^{35}+69$	$10^{35}+103$	10(31díg.)1720(30díg.)7107	0:00:00:00
$10^{50}+151$	$10^{50}+447$	1(47zeros)598(46zeros)7497	0:00:00:00
33 573 653	62 022 901	2 082 335 356 227 353	0:07:38:50
277 443 083	548 425 091	152 156 748 041 595 553	0:31:43:73
276 173 267	419 566 859	115 873 150 174 958 353	0:10:31:36
757 817 069	840 664 129	637 069 626 252 217 901	0:01:30:75
296 254 423	470 032 573	139 249 228 705 320 379	0:14:09:92
700 232 219	997 014 107	698 141 400 518 913 433	0:18:35:17

Taula 4.11

Resultats obtinguts a través del mètode de Fermat.

p	q	N	Temps
4 133	6 121 607	25 300 601 731	0:00:01:56
2 741	67 282 409	184 421 083 069	0:00:00:00
3	10 093	30 279	0:00:00:00
3	84 189 997	252 569 991	0:00:00:00
681 647	948 547	646 574 216 909	0:00:03:22
77 137	454 759	35 078 744 983	0:00:06:02
512 849	670 211	343 717 041 139	0:00:01:23
136 273	489 571	66 715 308 883	0:00:07:52
210 643	640 727	134 964 657 461	0:00:05:64
22 165 237	26 263 157	582 129 099 273 209	0:01:23:30
300 961	25 507 091	7 676 639 614 451	0:00:06:49
16 518 833	33 374 347	551 305 264 577 051	0:00:31:91
1 236 583	19 471 313	24 077 894 643 479	0:00:02:49
10 710 431	27 074 389	289 978 375 251 659	0:00:22:61
1	6 389 351	6 389 351	0:00:00:11
1	562 577	562 577	0:00:00:05
1	1 859 441	1 859 441	0:00:00:07
33 573 653	62 022 901	2 082 335 356 227 353	0:04:43:72
277 443 083	548 425 091	152 156 748 041 595 553	0:00:53:14
276 173 267	419 566 859	115 873 150 174 958 353	0:00:57:203
757 817 069	840 664 129	637 069 626 252 217 901	0:10:05:55
296 254 423	470 032 573	139 249 228 705 320 379	0:05:29:01
700 232 219	997 014 107	698 141 400 518 913 433	0:04:29:61

Taula 4.12

Resultats obtinguts a través del mètode de Lehman.

p	q	N	Temps
4 133	6 121 607	25 300 601 731	0:00:00:02
2 741	67 282 409	184 421 083 069	0:00:00:00
3	10 093	30 279	0:00:00:00
3	84 189 997	252 569 991	0:00:00:00
681 647	948 547	646 574 216 909	0:00:00:03
77 137	454 759	35 078 744 983	0:00:00:02
512 849	670 211	343 717 041 139	0:00:00:02
136 273	489 571	66 715 308 883	0:00:00:02
210 643	640 727	134 964 657 461	0:00:00:02
22 165 237	26 263 157	582 129 099 273 209	0:00:00:16
300 961	25 507 091	7 676 639 614 451	0:00:00:00
16 518 833	33 374 347	551 305 264 577 051	0:00:00:17
1 236 583	19 471 313	24 077 894 643 479	0:00:00:03
10 710 431	27 074 389	289 978 375 251 659	0:00:00:05
33 573 653	62 022 901	2 082 335 356 227 353	0:00:00:13
277 443 083	548 425 091	152 156 748 041 595 553	0:00:00:17
276 173 267	419 566 859	115 873 150 174 958 353	0:00:00:21
757 817 069	840 664 129	637 069 626 252 217 901	0:00:00:51
296 254 423	470 032 573	139 249 228 705 320 379	0:00:00:11
700 232 219	997 014 107	698 141 400 518 913 433	0:00:00:78
24829150509517	681832813845787	1692932955730453210331 3854879	0:02:48:70
71177928683	47603686631	338833181206919853 6973	0:00:10:31
784200405306371299	279333877966503023	219053740317132130117629324643936877	0:22:54:78

Taula 4.13

Resultats obtinguts a través del mètode rho de Pollard.

En el cas del mètode p-1 de Pollard finalment s'ha optat per una altra implementació ja que trobar el valor del nombre *B*-uniforme era un factor massa poc determinístic i a més requeria l'ús de logarítmes. Una interpretació del mateix algorisme observa que el $\mathrm{mcd}\left(a^k - 1, N\right)$ per un valor de k amb molts factors i provant valors diversos de a en el rang [2..*N*-2] acaba proporcionant un factor no trivial. Una expressió per k molt eficient és k=r! per la gran quantitat de divisors que té. Finalment es confecciona un algorsime que evalua $\mathrm{mcd}\left(a^{(g+j)!} - 1, N\right)$ que provi calors de a i de j fins a obtenir un factor no trivial, en l'entrada es necessita N i g. El que s'ha de definir amb molta cura és el valor de j perquè a vegades no s'obté un factor no trivial. Trigarà més amb aquells nombres que p-1 té menys divisors i p i q estan més separats.

p	*q*	*N*	Temps
4 133	6 121 607	25 300 601 731	0:00:00:03
2 741	67 282 409	184 421 083 069	0:00:04:94
3	10 093	30 279	0:00:00:00
3	84 189 997	252 569 991	0:00:00:00
681 647	948 547	646 574 216 909	0:00:04:95
77 137	454 759	35 078 744 983	0:00:47:34
512 849	670 211	343 717 041 139	0:00:04:89
136 273	489 571	66 715 308 883	0:00:05:32
210 643	640 727	134 964 657 461	0:03:05:13
22 165 237	26 263 157	582 129 099 273 209	0:03:15:68
300 961	25 507 091	7 676 639 614 451	0:00:00:31
16 518 833	33 374 347	551 305 264 577 051	0:12:05:36
1 236 583	19 471 313	24 077 894 643 479	0:05:23:13
10 710 431	27 074 389	289 978 375 251 659	0:08:37:53
882883		950 161 333 249	0:00:07:43
33 573 653	62 022 901	2 082 335 356 227 353	0:19:23:07
277 443 083	548 425 091	152 156 748 041 595 553	0:43:33:26
276 173 267	419 566 859	115 873 150 174 958 353	0:31:53:05
757 817 069	840 664 129	637 069 626 252 217 901	0:41:53:05
296 254 423	470 032 573	139 249 228 705 320 379	0:36:19:62
700 232 219	997 014 107	698 141 400 518 913 433	0:28:31:17

Taula 4.14

4. Conclusions

No existeix cap algorisme polinòmic per resoldre el problema de la factorització d'enters tot i que els diferents mètodes exponencials i subexponencials existents avui en dia resulten més o menys eficients en certs tipus de nombres. Després d'haver realitzat una experimentació amb els diferents mètodes de factorització es presenta un modest protocol a seguir a l'hora de factoritzar un enter.

El primer atac s'ha de dur a terme amb els mètodes exponencials de Pollard perquè tot i el seu caràcter poc heurístic són capaços de resoldre la majoria de nombres. En el cas del mètode rho és molt interessant provar diverses expressions del tipus $x^2 + a \pmod{N}$ perquè resulten sorprenentment eficients, factoritza nombres de 30 dígits en dos minuts! El mètode de Fermat com s'ha pogut constatar resulta altament eficient quan els dos factors són molt propers de manera que sempre és una opció a considerar. En cas que aquests intents fracassin s'executa el mètode de Lehman però cal remarcar que si es té accés a algun mètode subexponencial com el garbell quadràtic, *QS,* és perferible el seu ús.

5. Conclusions generals

Com que les conclusions de cada secció ja tenien un caràcter tècnic he decidit fer una reflexió de caire personal en les conclusions generals. En el treball de recerca he pogut observar com de la unió entre la programació i les matemàtiques més rigoroses en neix una disciplina que personalment trobo impressionant i molt atractiva. A partir de tots els coneixements teòrics apresos he estat en condicions de simular alguns dels problemes eternalment coneguts i més estretament vinculats als nombres primers.

Resulta sorprenent que donat un determinat problema, amb una sola solució correcta, puguin aparèixer quantitat de mètodes que el resolen amb major o menor eficiència i correcció. Realitzar la comparativa entre aquests i veure on realment deixen de ser competitius ha estat un procés molt dur i exigent però s'ha revelat altament gratificant quan es compara el temps del primer mètode amb el del darrer (que acostuma a ser el més complex).

Passar hores programant i després esperant que l'ordinador executés el que se li havia ensenyat a fer ha resultat molt enriquidor perquè ha relativitzat la meva concepció del temps i dels problemes; és a dir, quin sentit té resoldre un problema quan tota la matèria s'hagi col·lapsat en forats negres? Com a conclusió cal senyalar que es pot calcular $\pi(x)$ o enumerar els primers entre $[W, WL + B]$ amb mètodes que milloren $O(N / \log N)$ com el TNP suggeriria. Es pot verificar la primeritat en temps polinòmic en funció de la longitud de l'entrada reduint l'error a la mínima expressió. Finalment el problema de la factorització ha de ser considerat com intractable a causa de la seva complexitat subexponencial en els millors casos.

Els que creguin que aquest treball és excessivament teòric o que no té cap aplicació pràctica perquè tracta conceptes abstractes cal remetre'ls als apèndixs on quedarà demostrada l'actual relació entre privacitat i nombres primers.

6. Bibliografia

[1] Aho, A., Hopcraft, J. E., i Ullman, J. , *The Design and Analysis of Computer Algorithms*. Addison-Wesley Publications Company, 1974.

[2] Argrawal, M., Kayal, N., i Saxena, N. PRIMES is in P. Annals of Mathematics, 160 (2004): 781-793.

[3] Atkin, A. O. L. i Bernstein, D. Prime sieves using binary quadratic forms. Mathematics of Computation, 73 (2004): 1023-1030.

[4] Artjuhov, M., Certain criteria fort he primality of numbers connected with the Little Fermat Theorem. Acta Arithmeica, 12 (1966-67): 355-364.

[5] Bach, E. Explicit bounds for primality testing and related problems. Informatics and Computing, 90 (1990): 355-380.

[6] Bays, C., Hudson, R. Zeroes of Dirichlet L-functions and irregularities in the distribution of primes. Mathematics of Computation, 69 (2000):861-866.

[7] Bernstein, D. Detecting perfect powers in essentially linear time, Mathematics of Computation, 67 (1998): 1253-1283.

[8] Bernstein, D. Distinguishing prime numbers from composite numbers: The state of the art in 2004. http://cr.yp.to/primetests.html#prime2004

[9] Bohman, J. On the number of primes less than a given limit. BIT 12 (1972):576-588.

[10] Borwein, J., Bradley, D., i Crandall, R. Computational strategies for the Riemann Zeta Function. Journal of Computational and Applied Mathematics, 121 (2000):247-296.

[11] Crandall, R. i Pomerance, C. *Prime Numbers: A Computational Perspective*. Springer-Verlag 2005, segona edició.

[12] Cormen, T. H., Rivest, R. L., Leiserson, C. et al. *Introduction to algorithms*. MIT Press 2001, second edition

[13] Davenport, H. Multiplicative Number Theory. Springer-Verlag, 1980.

[14] Derbyshire, J. *Prime Obsession: Bernhard Riemann and the Greatest Unsolved Problem in Mathematics*. Plume 2004.

[15] Dickson, L. E. *History of the Theory of Numbers*. Volume I: Divisibility and Primality. Reimpressió: Dover Publications 2005. Capítol XVIII.

[16] Diffie, W. i Hellman, M.E. New directions in cryptography. IEEE Transactions on Information Theory, 22 (1976): 644-655.

[17] Edwards, H.M. *Riemann's Zeta Function.* Dover publications, 2001.

[18] ElGamal, T. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, 31 (1985): 469-472.

[19] Gauss, C. F. *Disquisitiones Arithmeticae.* Springer-Verlag, 1986

[21] Euler, L. *Variae Observationes circa Series Infinitas* publicat en *Commentarii academiae scientarum Petropolitanae* 9 (1737):160–188. Traduït a l'anglès per Pelegrí Viader, Lluís Biblioni i Pelegrí Viader (fill).

[22] Fortnow, L., Homer, S. *A Short History of Computational Complexity.* Butterworth Heinemann, 2006.

[23] Goldwasser, S. i Kilian, J. Almost all primes can be quickly certified. Procedures of eighteenth Annual ACM Symposium on Theory of Computing, (1986): 316–329.

[24] Hardy, G. H. *A Mathematician's Apology* . Cambridge University Press, 1940.

[25] Hardy, G.H. i Wright, J. *Theory of Numbers.* Oxford science publications, 1980.

[26] Havil, J. *Gamma: Exploring Euler's Constant*, Princeton University Press, 2003.

[27] Kahn, D. *The Codebreakers*, The MacMillian Company, 10 edició 1973.

[28] Knuth, D. *Seminumerical Algorithms: The Art of Computer Programming.* Volum 2(1981), segona edició.
[29] Knuth, D., Pardo, T. Analysis of a simple factorization algorithm. Theoretical Computer Science, 3 (1976-77): 321-348.

[30] Koblitz, N. *A Course in Number Theory and Cryptography.* Springer-Verlag, 1994.

[31] Koç, Ç. K. High-Speed RSA Implementation. RSA Laboratories, 1994.

[32] Lagarias, J. C., Miller, V.S., i Odlyzko, A.M. Computing $\pi(x)$: the Meissel-Lehmer Method. Mathematics of Computation, 44(1985):537–560.

[33] Lagarias, J. C. i Odlyzko, A. M., Computing $\pi(x)$: An analytic method, Journal of Algorithms, 8(1987), 173–191.

[34] Landau, E. *Handbuch der Lehre der Verteilung der Primzahlen.* University of Michigan Historical Math collection. Darrera visita 19/08/06
http://www.hti.umich.edu/cgi/t/text/text-idx?c=umhistmath;idno=ABV2766.0001.001

[35] Lehman, R. S. Factoring Large Integers. Mathematics of Computation, 126 (1974): 637-646.

[36] Lehmann, D.J. On primality tests. Journal of Computation, 11 (1982):374-375

[37] Lucena, J. *Criptografía y seguridad en computadores.* Criptored, 2002.

[38] Mapes, D. C. Fast method for computing the number of primes less than a given limit. Mathematics of Computation, 17 (1963): 179-185.

[39] Miller, G. L. Riemann's hypothesis and tests for primality. Journal of Computer System and Sciences, 13 (1976): 300-317.

[40] Miller-Rabin's Primality test, última consulta 11/12/06
http://www.security-labs.org/index.php3?page=5.

[41] Mills, W. A prime representing function. Bulletin of the American Mathematical Society, 53 (1947):604.

[42] Menezes, A.J, Oorschot, P. i Vanstone, S. A. *Handbook of Applied Cryptography*. CRC Press, 1995.

[43] Newman, D. J. *Analytic Number Theory*. Springer-Verlag, 2000

[44] Odlyzko, A. Tables of zeros of the Riemann zeta function. Última consulta 13/01/07 (ρ). http://www.dtc.umn.edu/~odlyzko/zeta_tables/index.html

[45] Rabin, M.O. Probabilistic algorithm for testing primality. Journal of Number Theory, 12 (1980): 128-138.

[46] Weisstein, E. W. Rabin-Miller Strong Pseudoprime Test. La última visita el 19/02/07.
http://mathworld.wolfram.com/Rabin-MillerStrongPseudoprimeTest.html

[47] Ribenboim, P. *The Little Book of Big Primes*. Springer-Verlag, 1991.

[48] Riemann, B. Ueber die Anzähl der Primzahlen unter einer gegebenen Grösse. Acadèmia de Berlin, 1859. Traducció a l'anglès de Wilkins. (On the number of prime numbers less than a given quantity).

[49] Riesel, H. i Göhl, G. Some calculations related to Riemann's prime number formula. Mathematics of computation, 24 (1970): 969-983.

[50] Rothe, J. *Complexity Theory and Cryptology: an introduction to cryptocomplexity*. Springer-Verlag 2005.

[51] Rivest, R.L., Shamir A., i Adleman, L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, 21(1978): 120-126.

[52] Rivat, J., Déléglise, M. Computing $\pi(x)$ the Meissel, Lehmer, Lagarias, Miller, Odlyzko Method. Mathematics of Computation, 65(1996):235-245.

[53] Sabbagh, K. *The Riemann Hypothesis: The Greatest Unsolved Problem in Mathematics*. Farrar, Strauss and Giroux, 2002.

[54 Sautoy, M. *The music of the primes: searching to solve the greatest mystery in mathematics*. Harper-Collins, 2003

[55] Schneier, B. *Applied Cryptography: Protocols, Algorithms and Source Code in C*, John Wiley & Sons, 1996.

[56] The Secretary of Defense (USA). *Comunications and Intelligence Activities.* The Secretary of State, 1952.

[57] Singh, S. G. *The code book.* Anchor Books, 2000.

[58] Solovay i Strassen. A fast Monte-Carlo test for primality. SIAM Journal on Computing, 6 (1977):74-86.

[59] Zimmermann, P. última vista el 14/02/07.
http://www.philzimmermann.com/EN/essays/WhyIWrotePGP.html

Nombres Primers: Apèndixs

34606293587437915183586403433802000335363774208015046139778344396288404561
63670274695558067710273338903291484649576746867361500680672973256849982294
17513471306046715315562916578073735806394397538783712970754922260336996128
05317953949314557472377971540956596986392946801273603806539268262325399274
03528251263161028965803081822842452827764247160796519008797451937040525226
08687857310011743423378122295942746674789456596942223207409763276892280138
83611312688579442713442836270044546362003259244360545075104863063226524790
48427184040269481583912267530925982537961**02**70118700367125393924605406339911
524194739457545185396447282023994090 7**0907**08493656143784 1**0765**69649640634474
1903269603648889963663737246994204 4**2554**863217246966**65332187**12507313639708 7
93251917881197671281119538022809 94**438**426044594 8**903899642 9**0757287248470096 8
1161322195343348876704409704218557**907**190837**578834354625**38672661002637144 18
30312162580651653360841240070839 57**76781**6**3822208244**6448262267622039393166 97
1691750638343530001877555467485239 39**41078203**252506454872266014618887853513
0403562196505238581180315698415257 3**793**024870297779017447665227647214954245
74672360807264818855365048175264**6217**3417675877031027841884390451703721233 7
126797678723265192297285462988**5275**805510799979966398278234291925904048451 9
916902662965916074408300912 9**8082**745165070462176222791200541201166213622619
058357232256320588683863 03**3349**2069531470667934520145523545715586807629775 7
50767336522244786318851 32**658**2406369353433842673596687146474559931436921303
381643378466193119754 59**739**148128917370732630093860114447625798664914715796
38840495533883290493 11**835**84176091992482881799743764781711694354498014995 43
6932255950515371200 12**254**28365862315993562543385568941336864496875135533510
60049968327337397 33**616**548919306680598358979928107396740463648225404401003 8
2350332409018410 69**580**669493263398105127420431884007797631055082279895020 80
1435295461738977 9**415**61387674459449971964118546877424106606477909182820092 0
130470633766365 9**956**63437027502646946214122756697764317533055320447917338 05
967601459296968 1**193**693848147978788200377582243177441076934032633202112736 5
10715705205176 7**507**4216411727293334350960532863677614082158400623689732266 4
92545395569414**228**89549827916454048032382606658732449665710218802075981945 0
849785966625 77**635**70712326151135268840204935893548466607645547452275906939 5
84836406254 29**1952**4288894620009970633830887635293472548456968258148058917 33
747597081723 6**3001**732337965475935340957514166563073128817366680609154447725
957926515002 6**66922**1584490458578744433996455125942074072284514345418787542 3
841522650568 5**870422776273844720530303396**1419857179899769472732379888473080
491451160793 9**8776187380520137584222487145 0**4170733730506334075782359008500 4
356078507397 18**04886170508055225080680256130**5793144202218495960668134932391
02771311781068 34**06665025926887648249115306 1**30328796712569586840667220180 93
5570107945656552276 47**67956**169708750151**52817**33463642593578580769487646983 67
30210264489290654490270152992933888645 0**386**968290826592764178310956135717 07
7135910947791010530746795494500636696 68**830**7697170978308677660772144685780 0
85019931037528506729090342588742271941**059**692507288253947483315367691425082
19966781401596439461164584543951926 95**581**073662502269548848826525866810545 5
61006080777421967373343756837125702 8**719**0743437002905310061629854748993216 2
873113340234334659358 00**17781**8907 3**1226**37851839360891507653613169025882459 64
82935865582649813670 6**91970459577002**437049151007140032716043264995770257122
01345525672024278338 8**2489009931 0**61611900282124853751745280804970460679842 3
0408465820691652933366 69**960**2978870786671988002 1[illegible]141
0560000000691778647261948849222819828311491035 8[illegible]483
0716651423897922388478524905599598338545062163 6[illegible]569
06544604015266014390412783873078827829418661589181967050673120870400000001
5511210043330985984047953

Introducció

En aquests apèndixs es mostren alguns altres aspectes en els què s'ha treballat però que després de la transició de criptografia a complexitat havíen quedat de banda i no es podien incloure en la memòria del treball pròpiament dita.

De manera que aquí es presenten dos apropaments a la criptografia, d'una banda un de més tècnic que dedica molt interès als mètodes de xifra de clau assimètrica, i per l'altra una perspectiva històrica. No es tracta de resseguir l'evolució dels mètodes de xifra al llarg de la història sinó prendre alguns exemples paradigmàtics del seu ús transcendental en la diplomàcia i el món militar.

Al acabar de llegir aquest breu tractat és important que el lector prengui consciència de la ubiqüitat de la cirptografia en la vida quotidiana, ja sigui en transaccions bancàries, converses telefòniques, emissions de radio i televisió... és a dir, la criptografia ha abandonat l'espionatge i forma part del dia a dia, és ella qui protegeix la nostra privacitat.

APÈNDIX A Criptologia

A.1 Introducció 1

A.1.1 Termes utilitzats i anglicismes....1
A.1.2 Criptologia com a ciència....1

A.2 Naixement de la clau pública 5

A.2.1 Intercanvi de clau Diffie-Hellman....5
A.2.2 RSA i Pretty Good Privacy....13
A.2.3 Signatura digital i criptoanàlisi de RSA....17
A.2.4 Expectatives de futur....20

APÈNDIX B: Transcendència històrica 22

B.1 **Època clàssica:** Xerxes i els grecs 24

B.2 **Edat Mitja i Renaixement:** la Reina Maria dels escocesos 25

B.3 **Època contemporània:** el Telegrama Zimmermann 39

Apèndix A

A.1 Introducció

A.1.1 Termes utilitzats i anglicismes

He trobat que paraules com: "encriptar", "desencriptar", no es troben reconegudes per l'Institut d'Estudis Catalans perquè són anglicismes, provenen dels verbs anglesos *encrypt* i *decrypt,* mentre sigui possible he intentat utilitzar les paraules que disposem en la nostra llengua per expressar les mateixes idees: en aquest cas *xifrar* i *desxifrar.*

Però cal centrar-se i fer una definició de *criptografia* que permeti avançar en la investigació:

A.1.2 La criptografia com a ciència

Si es va a buscar una definició de criptografia en un diccionari o enciclopèdia és molt probable que es trobin definicions similars a aquestes.

Defincició A.1.2.1 *Gran Enciclopèdia Catalana* :
"Art d'escriure en caràcters secrets. Hi ha tres tipus: les invisibles, fetes amb tintes simpàtiques, les convencionals, basades en l'ús d'elements convinguts que produeixen un efecte diferent de la comunicació real del text, i les xifrades, basades en la transposició de lletres, síl·labes o qualssevol element del text, o bé en llur substitució per altres, segons unes equivalències expressades en una taula o clau".

Definició A.1.2.2 *Diccionario de la Real Academia Española*
"arte de escribir mensajes con una clave secreta o de modo enigmático".

Precisions conceptuals La criptografia ha deixat de ser un art, s'ha convertit en una ciència estretament lligada amb les matemàtiques i la informàtica.

- Ja no només s'escriu sinó que es guarda en discs durs o bé CD's, fins i tot s'envia. Els arxius que es tracten no sempre són lletres poden ser de diversos tipus: ".gif, .jpg, .wav, .txt, .doc, .rar etc."

- No sempre s'utilitza una sola clau; a vegades una serveix per encriptar i una altra per desencriptar.

•La clau no té perquè ser secreta, hi ha alguns criptosistemes que són de clau pública que funcionen amb una gran enginy.

•Els missatges xifrats que es representen en sistema binari per als ordinadors no són enigmàtics.

•L'ús de tintes és molt minoritari ja que un cop detectades ofereixen molt poca protecció. Els caràcters no tenen perquè ser secrets: poden ser lletres de l'abecedari, però organitzades de manera inintel·ligible.

L'objectiu no és ni molt menys ridiculitzar aquests dos instruments de coneixe-ment tant valuosos com són l'*Enciclopèdia Catalana* o el *Diccionario de la RAE*, simplement pretenc posar de relleu que la criptografia és una ciència nova, que ha evolucionat molt ràpidament en les darreres dècades gràcies el progrés dels ordinadors, i que, per tant, certes definicions han quedat obsoletes.

Definició A.1.2.3 La criptografia és una branca inicial de les matemàtiques, en l'actualitat també avarca el camp de la informàtica i la telemàtica. Fa ús de mètodes i tècniques per protegir o xifrar un missatge o arxiu mitjançant un algorisme utilitzant una o més claus. Hi ha diversos tipus de criptosistemes que permeten assegurar els quatre aspectes fonamentals de la seguretat informàtica i la privacitat:

•Confidencialitat: l'objectiu és assegurar que el missatge només podrà ser llegit pel destinatari.

•Integritat: es tracta d'assegurar que el missatge no s'altera durant la transmissió i arriba al destinatari tal i com l'ha enviat el remitent.

•Autenticitat: Cal evitar la suplantació de la identitat. El destinatari ha de tenir l'absoluta certesa de saber qui és l'emissor. L'eina electrònica que substituirà la signatura manuscrita tradicional s'anomena signatura digital.

•No-repudiació: El remitent no pot negar haver enviat el missatge: és com dir que no es pot falsificar la signatura digital.

El fet que fins la dècada dels 80, els pocs llibres que hi havia escrits sobre la criptografia es trobaven en la secció d'arts ocultes de les biblioteques entre els d'astrologia i fetitxisme palesa la seva brevetat com a disciplina científica.

Axiomes criptogràfics A.1.2.4 (de Claude Shannon)

• Sempre hi haurà algun agent extern que si disposa de prou recursos serà capaç d'interceptar la comunicació i llegir el contingut del missatge.

• La màquina del secret perfecte funciona amb una clau infinita i aleatòria.

Durant la Guerra Freda, el telèfon vermell que connectava els despatxos dels presidents de la URSS i dels EUA utilitzava un mètode de xifra anomenat xifrat de Vernam que utilitzava com a clau una seqüència aleatòria i d'un sol ús, mai es xifrava dos cops amb la mateixa part de la clau. Les seqüències pseudoaleatòries s'obtenien a través de procediments físics o mecànics, consultant historials de loteries o casinos...

Aquesta disciplina no es preocupa en absolut de la seguretat física de la transmissió, es considera que sempre hi ha un usuari malintencionat capaç de llegir el missatge. Per tant, l'objectiu és alterar el missatge que s'envia a través del canal insegur, de manera que tan sols el destinatari sàpiga com reconstruir el text per extreure'n la informació.

Un criptosistema és un sistema criptogràfic però en general es fa referència a un conjunt d'algorismes necessaris per dur a terme el xifrat o desxifrat. Es classifiquen en dos grans grups: de clau pública i de clau privada. En tot criptosistema hi ha cinc elements molt importants: el missatge, el text xifrat, la clau, la transformació per xifrar i per desxifrar. Cal establir una notació clara i única per no crear ambigüitats al llarg del treball.

• M és un conjunt de missatges i un element de M és un text en clar.

• C és un conjunt de missatges xifrats i un element de C és un missatge xifrat.

• K és un conjunt de claus i un element de K és una clau.

• X és la funció o mètode de xifrat.

Si $k \in \{K\}$ llavors $X_k = M \to C$

• D és la funció o mètode de desxifrat.

Si $k \in \{K\}$ llavors $D_k = C \to M$

Per referir-nos a un element d'un conjunt el denotarem amb el símbol del conjunt al qual pertany però amb minúscula. Si e i d pertanyen a K, $X_e(M) = C$ i $D_d(C) = M$ només pot haver una d tal que $D_d(X_e(M)) = M$. Com es veurà més endavant hi ha casos en que d i e són iguals i d'altres on són diferents.

Definició A.1.2.5 Una funció f que relaciona dos conjunts X i Y s'anomena funció unidireccional si $f(x)$ és fàcil de calcular per tots els valors de $x \in \{X\}$ però és computacionalment impossible trobar la x a partir del valor y tot i conèixer $f(x)$. Les funcions unidireccionals amb trampa són aquelles que coneixent y, $f(x)$ i certa informació complementaria es pot realitzar el càlcul de x però que sense la informació extra resulta impossible.

El paradigma per excel·lència de funció unidireccional és la multiplicació d'enters. És fàcil calcular la multiplicació de dos enters, tot i que siguin grans; tanmateix fins i tot l'ordinador més potent i amb un algorisme eficient trigaria anys a factoritzar un nombre de tres-centes xifres producte de dos factors de similar longitud. El problema del logaritme discret és un altre exemple: mentre es pot calcular amb un temps raonable $x = a^z \bmod N$, el càlcul invers, és a dir, $z = \log_a x \pmod N$ comporta un volum de càlcul molt superior. N és el grup multiplicatiu finit d'ordre N i de generador a, on $a \in N$.

Aquest tipus de funcions són la base de la criptografia de clau pública. En principi $D_d = X_e^{-1}$ ha de ser computacionalment impossible de calcular per algú que tan sols coneix X i e. És a dir, si $f : M \to C$ és fàcil, ha de resultar impossible $f^{-1} : C \to M$. f és una funció unidireccional amb trampa perquè sense D_d no es pot obtenir M. Quan es diu que no té inversa és des d'un punt de vista computacional i no matemàtic, per tant, no les hem de confondre amb aquelles que no són un-a-un.

A.2 Naixement de la clau pública

A.2.1 Intercanvi de claus Diffie-Hellman

Durant la dècada dels 60 els ordinadors es van fer més potents i es van abaratir, de manera que, cada cop hi havia més empreses capaces de comprar-ne i utilitzar-los per a transaccions comercials i en alguns casos per xifrar importants operacions bancàries.

Amb la progressiva generalització de l'ús d'aquests per xifrar missatges van aparèixer nous problemes que no s'havien plantejat mentre la criptografia estava reservada a les esferes militars o diplomàtiques: la transformació de la criptografia d'art a ciència es va accelerar. Una de les primeres preocupacions era l'estandardització; si una empresa volia enviar un missatge a una altra havien d'acordar prèviament un sistema. El 15 de Maig de 1973 l'Oficina Nacional Americana d'Estàndards –*ANBS* són les seves sigles en anglès- volia solucionar aquest problema i va enviar una petició a les millors centres d'investigació per presentar un sistema de xifra estàndard per afavorir i facilitar les comunicacions i el comerç a distància.

D'entre els algorismes que es van presentar el que semblava més poderós era un producte d'IBM anomenat Lucifer. Havia estat programat per un immigrant alemany anomenat Horst Feistel, arribat el 1934, que havia tingut moltes dificultats amb l'administració americana pel seus orígens. Va haver de mantenir el seu interès per la criptologia en secret per evitar aixecar sospites; de fet la NSA –Agència de Seguretat Nacional-, el va obligar a interrompre les seves investigacions perquè volien mantenir el seu monopoli en la recerca criptogràfica. La NSA és una de les agències del govern nord-americà amb un pressupost de bilions de dòlars: compren més material informàtic i tenen a més matemàtics empleats que cap altra agència o empresa del món.

Feistel va acabar el seu projecte als laboratoris Thomas J. Watson d' IBM a Nova York a principis dels anys 70. Lucifer era un dels productes per xifrar més potents i va ser adoptat per moltes empreses, era inevitable que s'adoptés com a mètode estàndard de xifra, tanmateix la NSA va tornar a intervenir. La fortalesa de la xifra era tan gran que estava fora de les capacitats computacionals de la NSA per trencar-lo. L'agència no estava interessada en un mètode que fes impossible l'espionatge, tant necessari per la seguretat nacional especialment durant la guerra freda.

La fortalesa d'un mètode de xifrat depèn principalment del nombre possible de claus; un criptoanalista que vulgui desxifrar un missatge haurà de provar totes les claus fins que en trobi una que al descodificar reveli un text amb sentit. La NSA va fer pressió perquè es limités l'algorisme de Feistel en aquest sentit, reduint el nombre possible de claus a 10^{17}, assegurant així la privacitat i la seguretat a nivell civil però sabent que qualsevol missatge podia ser desxifrat en un temps raonable amb l'enorme capacitat computacional dels ordinadors de l'agència. D'aquesta manera es va acabar adoptant l'algorisme Lucifer el 23 de Novembre de 1976, i va ser anomenat DES- *Data Encryption Standard*, trenta anys més tard, encara moltes empreses utilitzaven aquest mètode tot i que ara el seu successor, l'IDEA es comença a imposar.

DES va resoldre el problema de l'estandardització, afavorint al comerç i mantenint la seguretat davant l'atac de qualsevol empresa rival perquè és computacionalment impossible trencar un missatge xifrat amb DES, tanmateix encara quedava problema per solucionar: la distribució de les claus.

Si una empresa ha d'enviar informació confidencial a un client i creu que la via telefònica pot ser insegura, llavors xifra el missatge m amb una clau k i envia el missatge per correu tot i que sigui insegur. El client, per desxifrar m, necessita una còpia del programa DES en el seu ordinador i la clau k, però aquesta clau no pot ser enviada per la mateixa via que m. L'única manera d'estar segur que la clau arriba al destinatari sense que ningú la pugui copiar o manipular és lliurar-la en persona, una operació cara en temps i diners. Als anys 70 els bancs, distribuïen les claus amb despatxos que portaven els empleats de confiança. Mentre el nombre de negocis creixia i s'havien d'enviar més missatges amb les corresponents claus, els bancs es van adonar que el mètode era inviable. El nombre de claus si N usuaris es volien comunicar era: $N(N-1)/2$

De fet el problema de la distribució de claus no era un problema que hagués aparegut llavors, durant la història tots els mètodes de xifrat, per potents que fossin, quedaven compromesos en el moment de l'intercanvi de claus: si dos individus es volien comunicar en secret havien de confiar en un tercer o bé desplaçar-se. Els estats invertien una enorme quantitat de diners en missatgers que portessin les claus perquè hi havia pocs missatges o bé perquè eren de gran importància.

Imaginem que l'Alicia vol enviar una carta molt personal a en Marc i sap que els empleats de correus no tenen escrúpols i obren totes les cartes, el problema és que l'Alicia i en Marc no es poden trobar per intercanviar una clau

perquè viuen en països diferents. L'Alicia col·loca el seu missatge en una caixa i la tanca amb un cadenat, guardant ella la clau; quan en Marc rep la caixa tancada no pot obrir-la però li col·loca un altra cadenat, guarda la seva clau, i reenvia la caixa. L'Alicia rep ara la caixa amb dos cadenats, n'obre un amb la seva clau, i torna a enviar la caixa a en Marc. Aquest cop però és diferent, només hi ha un cadenat i en Marc té la clau per obrir la caixa i llegir el missatge. L'Alicia i en Marc han pogut mantenir llur privacitat tot i comunicar-se per un canal insegur.

Les implicacions d'aquest exemple, que pot semblar trivial són enormes; es demostra que dos individus poden comunicar-se de manera segura sense necessitat d'intercanviar cap clau. Sovint el descobriment més genial és aquell més senzill, el que tothom veu però no s'hi fixa, o el més obvi però que mai ningú hi pensaria. De fet la història de la ciència n'està plena d'exemples com aquests, Newton i la Llei de la Gravitació universal de 1685: tothom sabia que les coses queien però no es preguntaven perquè. El pare de la genètica, Gregor Mendel, va fer unes deduccions genials a partir d'uns experiments senzills -però molt enginyosos- com el cultiu i creuament de diferents tipus de plantes de pèsol: molta gent havia cultivat pèsols, tanmateix no s'havien plantejat de què depenien els colors i la resta de característiques.

Tot i que sembla que el problema de la distribució de claus havia quedat solucionat, encara quedava un obstacle pràctic, el procés en el qual Alicia encripta, Marc xifra, Alicia decripta i Marc decripta. El problema és l'ordre en que l'encriptació i el desxiframent es duen a terme. Si l'última encriptació s'ha fet amb la clau k_I, el desxiframent s'ha de realitzar amb la clau del mateix usuari.

El problema de l'exemple anterior és que tot i que el Marc fa l'última fase d'encriptació, és l'Alicia qui desxifra primer, enlloc de ser en Marc qui ho fa. La importància de l'ordre del procés es veu clarament amb un exemple del dia a dia: al matí ens posem primer els mitjons i després les sabates, però a la nit ens traiem primer les sabates i després els mitjons: no ens podem treure els mitjons abans que les sabates!

L'aparició de la criptografia de clau pública és una de les revolucions més importants en criptografia, i de fet probablement sigui la única veritable. Fins llavors tots els mètodes criptogràfics es basaven en la idea de la transposició i la substitució d'una manera més o menys elaborada. Amb la disponibilitat dels ordinadors, capaços de realitzar càlculs a altes velocitats, va aparèixer el DES -de clau compartida-, considerat com la materialització dels esforços mil·lenaris per obtenir un mètode de xifrat segur.

La criptografia de clau pública desafia tota lògica, però aconsegueix resoldre el problema de la distribució de claus i la autenticació. També es coneix pel nom de criptografia asimètrica pel fet que s'utilitzen claus diferents per xifrar i per desxifrar, de manera que la de xifrat es pot fer pública sense comprometre la seguretat del criptosistema. La criptografia asimètrica es basa en funcions matemàtiques més que en substitució i permutació.

Aquest tipus de criptografia no és més segura -de per sí- des del punt de vista de resistència davant l'atac sistemàtic d'un criptoanalista ja que la seguretat depèn del volum computacional d'operacions que s'han de dur a terme per trencar el xifrat. L'aparició de la criptografia de clau pública no ha deixat obsolets els altres criptosistemes de clau privada perquè, a nivell de rapidesa, aquests darrers són molt més eficients.

Tanmateix, la criptografia de clau pública va solucionar els dos grans inconvenients dels sistemes de clau compartida com eren: la distribució o gestió de claus i l'autenticació. Els criptosistemes assimètrics han fet possible que dos usuaris puguin intercanviar informació confidencial, tot i comunicar-se per un canal insegur, sense haver de trobar-se prèviament per acordar una clau. El nombre de claus s'ha reduït considerablement de $N(N-1)/2$ a N.

Per fer-nos una idea d'aquest darrer aspecte; si considerem una xarxa de mil usuaris, necessitaríem mig milió de claus. Per una xarxa de 200 milions d'usuaris com el sistema telefònic Nord-Americà als anys 90, es necessitarien $2 \cdot 10^{17}$ claus, és a dir, 20 milions de bilions cada una d'aquestes lliurades a través de correus: inviable.

La criptografia de clau pública va néixer el maig del 1975 de la mà de dos investigadors del departament d'enginyeria electrònica de la Unviersitat de Standford: Whitfield Diffie i Martin Hellman. Van publicar el seu descobriment en un article anomenat *New directions in cryptography* poc abans de la Conferència Nacional d'Informàtica del 6 i 7 de Juny de 1976.

Withfield Diffie va néixer el 1944 i va passar la seva joventut en el barri novaiorquès de Queens, on se li va despertar l'interès per les matemàtiques llegint llibres com *Course of Pure Mathematics* de Godfrey Harold Hardy. Graduat el 1965 en matemàtiques al MIT, el 1970 ja era considerat com un dels majors experts en seguretat i, sigui dit, un dels pocs criptògrafs no empleats pel govern. Va adonar-se que si trobava una solució al problema de la distribució de claus

seria recordat per sempre com un gran criptògraf, raó per la qual es va posar a treballar intensament en el projecte.

Martin Hellman havia crescut crèixer en un barri jueu i després va mudar-se a un altre d'irlandés en el Bronx de Nova York on va començar a cultivar el seu interés per la criptografia des de molt petit per la seva dificultat a comunicar-se amb altres individus.

Diffie-Hellman van proposar un protocol d'intercanvi de claus de manera que dos usuaris que no s'haguessin pogut trobar mai per acordar una clau K, comunicant-se per un canal insegur i intercanviant dades no-confidencials, poguessin acordar-ne una que tan sols ells dos poguessin saber. Els detalls els explicarem tot seguit, però fins i tot si tota la comunicació havia estat seguida per d'altres ells eren els únics capaços de calcular K. Però aquest protocol no permetia xifrar informació de manera que el seu ús no es va generalitzar.

Si bé és cert que la seva idea és brillant i d'un enginy sorprenent, hi havia hagut altres experts investigant el problema de la distribució i l'autenticació com: Horst Feistel –dissenyador del DES- o Ralph Merkle i en el descobriment de la criptografia de clau pública Diffie-Hellman van considerar idees de col·laboradors com John Gill, enginyer electrònic que va suggerir l'exponenciació modular o Donald Ervin Knuth, professor d'Informàtica de la universitat Californiana Standford i que va donar la idea de treballar amb nombres primers i la seva difícil factorització. Per tant, podríem afirmar que el naixement de la criptografia de clau pública no només va ser gràcies a Diffie-Hellman –si bé varen tenir un paper fonamental- sinó que va ser fruit de les idees aportades per diversos experts.

El 1975, tres matemàtics dels GCHQ (Government Communications Headquarters) britànics havien descobert la criptografia de clau pública. James Ellis, Clifford Cocks i Malcolm Williamson però per raons d'espionatge van restar a la ombra i el seu descobriment només va ser utilitzat pels serveis d'intel·ligència durant un breu període. Els informes d'aquests es van desclassificars a finals de la dècada dels 90.

El concepte de la clau pública està molt més enfocat a una xarxa d'usuaris que no a una parella. És una mostra de l'adaptació de les eines criptogràfiques a les necessitats modernes; amb l'aparició del *Internet* la necessitat de mantenir les comunicacions secretes havia sortit definitivament del àmbit militar o diplomàtic per entrar de ple en el dia a dia de moltes empreses i ciutadans.

Cada usuari té dues claus K_p (pública) i K (secreta); mentre K_p es publica en un directori públic perquè tothom hi tingui accés, guarda K i només ell la coneix. Els nombres utilitzats com han de complir unes característiques molt estrictes -que es veurà més endavant- i es generen mitjançant un algorisme de generació de claus.

Un usuari que vulgui enviar un missatge M a u ha de buscar K_p^u en el directori públic, i tot seguit aplicar l'algorisme de xifratge públic: C=X(K_p^u,M). El criptograma s'envia a l'usuari u i tot seguit ell aplica l'algorisme de desxifratge públic amb la seva clau K_s^u i obté M=D(K_s^u,C).

Amb aquest mètode queda reduït el nombre de claus, de $n(n\text{-}1)/2$ a n. Cal que M= $D(K_s^u, X(K_p^u, M))$ i per tant s'ha d'anar amb compte amb l'ordre en què es realitzen les fases de xifra i desxifrat.

En la seva publicació Diffie i Hellman, presenten diverses perspectives d'investigació per poder implementar un criptosistema viable de clau pública amb les funcions unidireccionals. Per aconseguir que sigui computacionalment impossible obtenir M a partir de C sense conèixer K_s^u Diffie i Hellman presenten la exponenciació modular com la millor una funció unidireccional amb trampa molt útil. Això es deu al fet que el càlcul del logaritme discret amb el millor algorisme triga aproximadament $\sqrt{N}$ operacions, on N és el mòdul de la exponenciació. Actualment el millor algorisme funciona amb una complexitat de:

$$O = e^{\sqrt{\ln N \cdot \ln \ln N}}$$

de manera que el volum computacional no ha canviat significativament.

Utilitzar l'àlgebra lineal també va ser una de les idees que van tenir en compte ja que calcular la inversa d'una matriu és molt més difícil que obtenir un parell de matrius inverses qualssevol; tanmateix finalment van adoptar l'aritmètica modular ja que satisfeia millor les seves necessitats.

Les funcions basades en l'aritmètica modular tenen un comportament força erràtic, i per a un criptoanalista li és molt més difícil ja que no pot utilitzar mètodes aproximatiu.

Whitfield Diffie i Martin Hellman van proposar al Congrés Nacional d'Informàtica dels Estats Units un protocol d'intercanvi de claus que permetia a dos usuaris acordar una clau comunicant-se a través d'un canal insegur.

1- Els usuaris i i j acorden un generador a i un grup multiplicatiu d'ordre N on $a \in N$.

2- Usuari i tria un nombre aleatori X_i dins N i el guarda en secret. Però publica Y_i, $Y_i = a^{X_i} \bmod N$.

3- Quan l'usuari i es vol comunicar amb j calcula $K_{ij} = a^{X_i X_j} \bmod N$, com no coneix X_j pren $K_{ij} = (Y_j)^{X_i} \bmod N$ on Y_j és la clau pública de j.

Al final del protocol i i j han pactat un element $a^{X_i X_j}$ en el mòdul n i secret per la resta d'usuaris. La seguretat del protocol d'intercanvi de Diffie-Hellman resideix en la dificultat de calcular logaritmes, és a dir si un usuari volgués calcular K_{ij} a partir de Y_i i Y_j hauria de realitzar: $K_{ij} = Y_i^{(\log_a Y_j)} \bmod N$. No s'ha trobat cap altra manera de calcular K_{ij} sense obtenir prèviament X_i o X_j. Si representem n, X_i i X_j en binari, és a dir, com 2^v, on v és la llargada de bits, el nombre de multiplicacions per realitzar la exponenciació mod n serà aproximadament $2v$.

La longitud de les claus que s'acostuma a recomanar actualment és de 1024 bytes, és a dir 2^{1024} (10^{308}) en decimal a tot estirar, prendria 2048 multiplicacions. Mentre que calcular $K_{ij} = Y_i^{(\log_a Y_j)} (\bmod N)$, és a dir, $\log_a Y_j \bmod N$, pren $\sqrt{2^{1024}}$ o 2^{512}. La grandesa d'aquest protocol és la seva aparent simplicitat, els usuaris i i j intercanvien: n, Y_i, Y_j, a través d'un canal insegur sense que ningú excepte ells pugui obtenir K_{ij} ($a^{X_i X_j} (\bmod N)$) amb un temps raonable.

La autenticació és un dels objectius més importants a l'hora de dissenyar un criptosistema. Ha de ser fàcil per tothom reconèixer que una signatura és autèntica però impossible per a ningú altre de produir-la. Si un banc rep l'ordre d'un hipotètic client que s'ha de dur a terme una transferència del compte A al B, la institució necessita saber que aquest usuari és efectivament el propietari del compte corrent.

Els protocols d'autenticació estan dissenyats perquè dos usuaris comunicant-se a través d'un canal insegur puguin mantenir: la confidencialitat, la no repudiació, la autenticació del missatge i de l'usuari i mantenir la integritat del missatge.

La criptografia de clau pública permet la ràpida identificació dels usuaris mitjançant el procés següent. Prenem dos usuaris, l'Alícia(A) i el Bernat(B), on X_k és el procés de xifra amb la clau k. Quan l'Alícia vol enviar un missatge al Bernat transmet $X_b(M)$ on b és la clau pública de B, però la diferència està en què inclou al principi (o al final) un C obtingut a partir de $X_b X_a^{-1}(M)$. On X_a^{-1} és la inversa de X_a només calculable si es té la clau privada de A, i aquesta només la té l'Alícia.

Quan el Bernat apliqui $X_b^{-1} X_b(M)$ amb la seva clau privada obtindrà el missatge M, i un tros sense cap sentit corresponent a la part xifrada amb la clau privada de l'Alícia. Si en aquesta part aplica $X_a X_a^{-1}(M)$ amb la clau pública d'Alícia obtindrà un altra missatge M, amb sentit que detallarà qui és l'autor.

A.2.2 RSA i Pretty Good Privacy

Ronald Rivest, Adi Shamir i Leonard Adleman van fer la major contribució que mai ningú ha fet a la criptografia de clau pública -després de la seva aparició-. Després de llegir l'article de Diffie i Hellman de 1976, van començar una recerca al Massachusetts Institute of Technology per dissenyar un criptosistema de clau pública funcional.

Rivest és un informàtic amb una facilitat sorprenent per absorbir idees, aplicar nous conceptes en llocs impensables i sempre coneixedor de l'actualitat científica. Shamir també és un informàtic, únic a l'hora d'afrontar problemes, i Adleman és un matemàtic amb un gran rigor a l'hora de procedir i afrontar problemes. Durant diversos mesos, Rivest i Shamir proposaven noves idees mentre Adleman en buscava els defectes; no va ser fins al Maig de 1977 que el seu esforç es va veure recompensat i van trobar la manera d'implementar un criptosistema de clau pública.

Tot i que des del seu naixement el 1976 la criptografia de clau pública havia gaudit d'un cert prestigi gràcies al seguiment mediàtic que se'n va fer entre la premsa científica, les institucions oficials com el National Bureau of Standards o la National Security Agency es van decantar per un altra criptosistema com a estàndard, el ja mencionat, DES. És cert que si bé, el 1976 no existia cap criptosistema basat en les idees de Diffie i Hellman, la NSA no va dedicar cap esforç en el disseny d'un mètode de xifrat de clau pública i va afirmar que deu anys abans ja havien descobert la criptografia assimètrica sense aportar-ne proves.

La gran feblesa de la criptografia de clau pública és la velocitat d'execució, el DES arriba a funcionar mil vegades més ràpid: les implementacions de RSA més eficients xifren uns milers de bits per segon, mentre que el DES a milions per segon. És per aquest motiu que en molts casos s'utilitzen sistemes *híbrids*, on el criptosistema de clau pública s'utilitza tan sols per acordar una clau.

Un dels primers defensors incondicionals de RSA va ser Gustavus Simmons, cap del departament de matemàtiques d'uns laboratoris estatals, i encarregat dels aspectes matemàtics del control de les armes nuclears. Les signatures digitals que proporcionava RSA li serien molt útils per: assegurar que les ordres d'armar la bomba provenien d'un emissor legítim i per evitar que els informes sísmics fossin manipulats pels intermediaris.

En plena Guerra Freda, després d'haver signat el 1963 el Tractat de Limitació de Proves Nuclears, els EUA havien col·locat sismògrafs en els països aliats per controlar el nombre i la potència de les proves que realitzava l'enemic. RSA permetia xifrar les dades de manera que no poguessin ser alterades però tanmateix el país "hoste" podia llegir sense manipular les dades que sortien de les seves fronteres.

L'RSA és un criptosistema assimètric i de bloc: funciona amb dues claus diferents, una per xifrar i l'altra per desxifrar. La funció unidireccional amb trampa utilitzada en aquest mètode es basa en el problema de la factorització d'enters i el logaritme discret. Les claus en RSA es generen seguint la idea de Diffie-Hellman de la clau pública i per tant desapareix la necessitat de correus per transportar les claus i existeix la possibilitat de signar digitalment els missatges; de manera que, els missatges seran privats i estaran firmats.

Cada usuari u tria dos nombre primers: p i q, i calcula $n=p\cdot q$. Per tant el grup multiplicatiu que utilitzarà serà $\mathbb{Z}^*_N$, és a dir, a cada exponenciació haurà de prendre el resultat mòdul n. L'ordre del grup multiplicatiu serà –tenint en compte que n és producte de primers- $\phi(N)=\phi(p)\phi(q)=(p-1)(q-1)$.

U pren un enter positiu e, que compleixi $mcd(e,\phi(N))=1$ i $1\le e<\phi(N)$. Ara calcula d, que és l'invers multiplicatiu de e en $\mathbb{Z}^*_{\phi(N)}$, i obtindrem que $ed \bmod \phi(N)=1$. U ja disposa de la clau pública e i la clau privada d. D, p, q i $\phi(N)$ s'han de mantenir en secret.

Per xifrar un missatge M es converteixen els caràcters en nombres seguint el codi ASCII (American Standard Code for Information Interchange), tot seguint es divideix M en bocins. Cada un d'aquests trossos tindrà una representació binària $2^r < N \le 2^{r+1}$, direm per tant que RSA és un xifrador en bloc.

Quan A vol enviar un missatge M a l'usuari B, prendrà la clau pública de B, e_b, per calcular $C = M^{e_b} \bmod N_b$ i enviar-ho. Tot seguit B per obtenir M realitza les següents operacions:

$$C^{d_b} \bmod N_b = \left(M^{e_b}\right)^{d_b} = M^{e_b d_b} \bmod N_b$$

Hem de demostrar que $M^{ed} \equiv M \bmod N_b$. Utilitzant un teorema d'Euler segons el qual: donats dos nombres primers, p i q, i dos enters, N i M, tal que $N=pq$ i $N > M > 0$; per qualsevol enter k es compleix la següent relació:

$$x^{1+k\phi(N)} \equiv x \pmod{N}$$

que és una generalització de

$$x^{\phi(N)} \equiv 1 \pmod{N}$$

i retornant a la demostració,
on x és M i n el mòdul de l'usari B, N_b

$$M^{1+k\phi(N_b)} \equiv M \bmod N_b$$
$$M^{1+k(p-1)(q-1)} \equiv M \bmod N_b$$
$$\text{si} \quad ed = k\phi(N_b) + 1$$
$$\text{llavors } ed \equiv 1 \bmod \phi(N_b)$$
$$d \equiv \frac{1}{e} \bmod \phi(N_b)$$

Per tant d i e són inversos multiplicatius en $\mathbb{Z}^*_{\phi(N_b)}$. Retornem però al procés de desxifrat:

$$C^{d_b} \bmod N_b \equiv \left(M^{e_b}\right)^{d_b} \equiv M^{e_b d_b} \bmod N_b$$
$$M^{1+k\phi(N)} \bmod N_b = (M \bmod N_b)(M^{\phi(N_b)} \bmod N_b)^k \bmod N_b = M$$

Un criptoanalista que vulgui trencar el criptosistema RSA haurà de trobar d; i la única manera de trobar aquesta és a través del càlcul de l'invers multiplicatiu mòdul:

$$d = inv\left(e, \bmod \phi(N)\right) = e^{-1} \left(\bmod \phi(N)\right)$$

Després de 20 anys no s'ha trobat cap altra mètode eficient de càlcul de l'invers que no passi per trobar la factorització de N. Per tant s'han inveritt temps i esforços en el disseny d'algorismes el seu anàlisi entre els quals destaquen *QS, NFS, SNFS* i el mètode de corbes el lítptíques.

Especialment per RSA, les claus utilitzades s'han d'obtenir a través d'un algorisme amb unes característiques molt concretes per dificultar al màxim la tasca del criptoanalista quan intenti factoritzar n en pq. Quan l'usuari i triï els primers p i q per generar el seu grup multiplicatiu, hauran de complir les següents característiques:

1) p i q han de ser d'una longitud similar però no massa propers. Si p i q fossin propers per $p > q$,l'$\sqrt{N}$ seria més petit que$(p+q)/2$; i $(p-q)/2$ seria petit.

Tenint en compte que

$$\frac{p^2+q^2}{2^2}+\frac{N}{2}-N=\frac{p^2+q^2}{2^2}-\frac{N}{2};$$

el punt clau és veure que $\frac{2pq}{2^2}=\frac{N}{2}$

$$\frac{(p+q)^2}{2^2}-N=\frac{(p-q)^2}{2^2}$$

Si $a=(p+q)/2$ i $b=(p-q)/2$, llavors $p=a+b$ i $q=a-b$. Per factoritzar n haurem de provar valors de a majors que $\sqrt{N}$ fins que $a^2-n=b^2$, on b és un quadrat perfecte.

2) Els factors de $(p-1)(q-1)$ han de ser grans perquè sinó mètodes com els de Pollard resulten molt efectius i una clara amenaça a la seguretat.

3)El mcd(p-1,q-1) ha de ser petit. Hem de tenir en compte que p-1 i q-1 són parells i per tant divisibles per 2^2. El procés de desxifrar es du a terme amb $d=\text{inv}(e,\phi(N))=inv(e,\text{mcd}(p-1,q-1))$. Si el $\text{mcd}(p-1,q-1)$ fos gran llavors el $\text{mcd}(p-1,q-1)$ seria petit i per tant d es podria calcular com $\text{inv}(e,\text{mcm}(p-1,q-1))$ de manera més fàcil.

Finalment la NSA, davant la incapacitat de crear un centre de distribució de claus que pogués respondre a les necessitats en seguretat i privacitat dels seus ciutadans i govern, va adoptar la criptografia de clau pública per l'intercanvi de claus.

L'únic cas en què la criptografia de clau pública es podria veure obsoleta o fora de lloc, seria si es produís una veritable revolució en el càlcul de logaritmes discrets o bé en la factorització, o en cas que la teoria numèrica trobés un mètode diferent pel càlcul de l'invers multiplicatiu. Quan parlem d' un criptosistema de clau pública quasi sempre fem referència a RSA perquè s'ha implantat com l'únic veritablement segur després que d'altres com el d'El Gammal hagin quedat relegats a un ús molt minoritari.

A.2.3 Signatura digital i criptoanàlisi de RSA

En l'era on les transaccions i els correus electrònics, la necessitat de dissenyar un mètode que permeti signar els missatges esdevé cabdal. Signar és una qualitat que té més valor que autenticar ja que no només permet al recipient de verificar qui és l'emissor- autenticar- sinó que el recipient pot convèncer una tercera persona que només l'emissor ha pogut enviar aquest missatge, d'aquesta manera s'assegura la no-repudiació.

Una signatura electrònica ha de ser dependent del missatge i del signant, per evitar que el recipient pugui manipular el missatge, sigui canviar el contingut o la signatura, abans de mostrar el missatge a un hipotètic jutge. Quan el Bernat vulgui enviar un missatge a l'Alícia signat haurà de calcular la signatura S del missatge M utilitzant la seva clau de dexifrar privada D_B i tot seguit xifrar la signatura amb la clau pública de l'Alícia.

$$S_B = D_B(M)$$

Recordem que en un criptosistema assimètric la transformació $X(D(M)) = M$, i per tant no altera el missatge. Si generalitzem la idea de fet cada missatge M en el fons és un C d'un altre missatge.

Quan l'Alícia rep el missatge signat M primerament desxifra amb la seva clau privada, D_A, i tot seguit xifra el contingut amb la clau pública del Bernat, X_B. Per tant $X_B(D_A(X_A(D_B(M)))) = M$. Es pot observar que el procés de signar, xifrar i desxifrar és llarg, les millors implementacions de RSA arriben a ser unes 1000 vegades més lentes que les de DES, el criptosistema simètric de la NBS. Rivest, Shamir i Adleman ja n'eren conscients que el seu mètode era més lent i que o bé requeria un *hardware* especial o bé s'havia de restringir el seu ús.

És evident que l'arquitectura d'aquests criptosistemes té unes avantatges que altres mètodes no proporcionen com és la capacitat de signar els missatge i solucionen el problema de la distribució de claus. S'han dissenyat criptosisitemes híbrids que incorporen les avantatges de tots dos tipus d'estructura.

Quan volem xifrar el missatge, $M^X \bmod n$, apliquem el mètode de l'exponen-ciació modular de manera que reduirem les $X-1$ multiplicacions a $2 \cdot \log_2(X)$ multiplicacions i $2 \cdot \log_2(X)$ divisions. Els nombres primers juguen un paper fonamental en aquest criptosistema de clau pública.

No existeix cap mètode per provar matemàticament que un mètode de xifrat és segur; la única manera de verificar-ho és fer públic l'algorisme per que la comunitat científica pugui realitzar atacs a la seguretat. Si passat cert temps el mètode resisteix, llavors pot ser adoptat per aplicacions pràctiques. El DES va ser posat a prova durant disset anys a IBM abans de ser adoptat oficialment.

Per trencar RSA el criptoanalista haurà de factoritzar N. Si intenta calcular l'invers multiplicatiu de e mòdul N a través de $\phi(N)$, inevitablement necessitarà els factors p i q. Hi ha hagut molts esforços per obtenir $\phi(N)$ sense factoritzar N però $\phi(N) = (p-1)(q-1) = pq - p - q + 1 = n - (p+q) + 1$ i calcular $\phi(N)$ no és més fàcil que factoritzar N. El càlcul d'arrels de e-grau mòdul N podria ser un altra apropament però sembla que computacionalment és igualment intractable.

La seguretat del criptosistema RSA resideix en la dificultat computacional de factoritzar n, i és que des de la seva implantació els anys el 1978 encara ningú no ha pogut demostrar que es pugui obtenir d sense fer el càlcul de $d = \text{inv}(e, \phi(N))$.

L'agost de 1977 Martin Gardner, en la seva columna anomenada "Jocs Matemàtics" del *Scientific American*, va presentar la criptografia de clau pública i va reptar als seus lectors que desxifressin un criptograma obtingut a partir de la clau e en mòdul N.

> N=114 381 625 757 888 867 669 235 779 976 146 612 010 218 296 721 242 362 562 561 842 935 706 935 245 733 897 830 597 123 563 958 705 058 989 075 147 599 290 026 879 543 541

La recompensa eren 100$ i es van trigar 17 anys en desxifrar el missatge. El 26 d'Abril de 1994, una coordinació de 600 voluntaris i 1600 CPU's durant sis mesos va permetre dur a terme el desxiframent. Pot semblar que 6 mesos és poc temps però s'ha de tenir en compte que el nombre utilitzat té una magnitud al voltant de 10^{130}, mentre que ara s'utilitzen nombres de 2048 bits és a dir al voltant 10^{620}. Es van trigar 15 anys en disposar del poder computacional en ordinadors domèstics per poder realitzar un atac viable en temps.

> q=3 490 529 510 847 650 949 147 849 619 903 898 133 417 764 638 493 387 843 990 820 577
> p=32 769 132 993 266 709 549 961 988 190 834 461 413 177 642 967 992 942 539 798 288 533

M= "*The Magic Words are Squeamish Ossifrage*"

Ossifrage fa referència a la pràctica del voltor de carronyes, anomenat "Trencalós", de deixar caure els ossos de les seves preses des de gran altura per trencar-los i extreure'n així el moll de l'os. Utilitzar la frase -*M*- s'ha convertit en una pràctica força comuna, quasi de tradició, en molts reptes criptogràfics.

La importància d'aquest repte, més enllà del fet merament anecdòtic o històric, resideix en veure la cursa que s'ha establert entre la longitud de claus, és a dir el volum computacional que s'ha de dur a terme, i la capacitat dels ordinadors per resoldre aquest volum de càlcul en un temps raonable.

Recordem que la informació té valor mentre a algú li interessi. De manera que, per assegurar la seguretat del criptosistema, però en definitiva de la nostra comunicació; cal establir una longitud de clau superior a la capacitat de qualsevol grup d'ordinadors (almenys domèstics). El grau de seguretat és proporcional a la longitud de la clau. Ronald Rivest va estimar l'any 1977 que es trigarien 40 quadrilions - $4 \cdot 10^{16}$ - d'anys en factoritzar n i per tant desxifrar M.

La expansió del sistema RSA i l'intercanvi de claus exponencial en un mòdul N ha portat a un important desenvolupament de la multiplicació, la factorització i la busca de grans nombres primers. Fins ara, els atacs del RSA porten a la factorització de manera que aquest tipus d'algorismes han anat millorant la seva eficiència. Davant aquest fet per mantenir la seguretat s'ha hagut de treballar en un mòdul major, de manera que, diferents algorismes per buscar primers grans s'han anat succeint.

L'aritmètica de nombres grans i la teoria numèrica han estat molt estudiades però especialment la multiplicació ja que tant la divisió com la exponenciació o la factorització passen per trobar algorismes de multiplicació eficients.

La capacitat de càlcul es mesura en MIPS-any, és a dir, milions d'instruccions per segon que pot dur a terme un processador funcionant ininterrompudament durant un any. Un Pentium a 200-MHz processava uns 50-MIPS mentre que actualment, un Pentium IV a 2.4 GHz processa al voltant de 700-MIPS. Però no es tracta tan sols de la màquina que du a terme les operacions, sinó de les instruccions que li són donades per resoldre un determinat problema, és a dir, l'algorisme.

A principis dels anys 90 s'utilitzava un algorisme força eficient, anomenat garbell quadràtic -*Quadratic Sieve*- que era capaç de factoritzar un nombre duent a terme només un 10% de les operacions. Més tard, es va introduir un nou algorisme anomenat garbell de nombres especials -*Special Number Field Sieve*- o *SNFS*.

Si bé és cert que a nivell conceptual el criptosistema RSA és segur i es pot anar adaptant la longitud de les claus segons les necessitats de seguretat i el progrés dels ordinadors o dels algorismes de factorització; queda sempre la possibilitat que RSA sigui trencat analíticament, és a dir, es descobreixi algun mètode per calcular $\phi(N)$ sense factoritzar.

A.2.4 Febleses i expectatives de futur

Un criptoanalista pot aprofitar algunes vulnerabilitats es produeixen quan el valor de X no compleix unes característiques determinades o quan es firma de manera incorrecta.

Una clau dèbil és una X que quan aplicada al missatge M el deixa inalterat, cal considerar que sempre hi ha alguna possibilitat que això succeeixi.

$$M^X \equiv M (\mathrm{mod}\, n)$$

De fet, sempre hi haurà missatges que quedaran inalterats sigui quin sigui el mòdul N, per tant, la nostra tasca serà reduir al màxim el nombre de casos i per tant la probabi-itat d'enviar un missatge sense xifrar. Sigui $n = p \cdot q$ i X la clau privada per xifrar:

$$\delta_n = [1 + mcd(X-1, p-1)] \cdot [1 + mcd(e-1, q-1)]$$

δ_n denota el nombre de missatges que quedaran inalterats en mòdul N. Si $p = 2p'+1$ i $q = 2q'+1$, per p' i q' primers, llavors $mcd(X-1, p-1)$ pot valer 1, 2 o p' i el mateix passa per q. Llavors els valors de δ_n poden ser 4, 6, 9, $2(p'+1)$, $2(q'+1)$, $3(p'+1)$, $3(q'+1)$ o $(p'-1)(q'-1)$. Si volem evitar aquests casos el que hem de fer és calcular δ_n després de generar el parell de claus.

L'atac de l'intermediari és molt interessant, pot ser evitat si es prenen les mesures de seguretat necessàries. Imaginem que el Bernat vol enviar un missatge a l'Alícia i la Carles espia la comunicació. Quan el Bernat sol·liciti la clau pública a l'Alícia i aquesta li enviï; el Carles intervindrà la comunicació prenent X_A i

enviant al Bernat X_C, una clau pública generada per ell. Bernat xifrarà el seu missatge M amb X_C, el Carles la desxifrarà amb la seva D_C i enviarà a l'Alícia el missatge xifrar amb la clau pública correcte-X_A-.

La situació presentada és compromesa, hi ha dos entitats que creuen estar mantenint una comunicació segura quan hi ha un tercer membre que secretament intervé tots els contactes. Per evitar aquesta situació, la única sortida és que la clau pública estigui firmada per un tercer individu de confiança que certifiqui l'autenticitat.

En un article d' Adi Shamir i Eran Tromer publicat el 23 de gener del 2003, descrivien com dissenyar una màquina anomenada *TWIRL* capaç de realitzar factoritzacions aprofitant un hardware concret i les darreres innovacions en algorismes de factorització, el conegut *NFS* . La seva recerca en l'Institut Weizmann de Ciència d'Israel, van veure que la part computa-cionalment més costosa era el procés de garbell. La màquina TWIRL amb un cost inicial d'un milió de dòlars, amb un sobrecost de 10 milions de dòlars és capaç de fer el garbell d'una clau de 1024 bits en 1 any i amb un afegit de 10,000 dòlars al preu inicial triga 10 minuts amb una clau de 512 bits. Aquesta innovació pot aixecar preocupació sobre la longitud necessària de les claus.

És obvi però que no totes les agències o empreses tenen un pressupost que els permeti construir una màquina *TWIRL* i el seu ús quedaria possiblement restringit a agències estatals de seguretat o espionatge. Tanmateix l'empresa que comercialitza RSA ja ha presentat un repte per veure qui és capaç de factoritzar una clau de 2048-bits amb un premi de $200 000. És probable que veiem que les claus de 1024 o de 2048 bits no poden protegir la informació més que uns pocs anys davant d'un atacant amb recursos.

Apèndix B

Durant milers d'anys civilitzacions senceres han depès de la seguretat en la comunicació per governar i comandar els seus exèrcits. Eren conscients que si els seus missatges queien en mans enemigues, la informació es giraria en contra d'ells donant una informació molt valuosa a estats rivals. Era aquesta amenaça la que va motivar el desenvolupament dels codis i els xifrats: mètodes per enviar missatges de manera encoberta per tal que només el receptor desitjat fos capaç de llegir-los i entendre'ls.

La disciplina ancestral de la criptografia ha protagonitzat fets històrics relacionats amb la diplomàcia, l'espionatge i la caiguda i el ressorgiment d'imperis o civilitzacions. Per aquest motiu, els estats van desenvolupar departaments dedicats al xifrat de missatges i a la recerca dels millors mètodes per mantenir la seguretat de les comunicacions. Al mateix temps els trencaxifrats enemics destinaven gran quantitat de temps i esforç per trencar descodificar els missatges xifrats. Els trencaxifrats eren alquimistes lingüístics que intentaven extreure paraules amb sentit d'un conjunt de símbols. La història dels codis i els xifrats és la història d'una guerra entre xifradors i trencacodis, una raça d'intel·lectuals que va tenir un gran impacte en els curs de la història.

Aquest Apèndix B del treball ha estat concebut per posar de relleu l'enorme transcendència històrica de la criptografia i no per estudiar la evolució dels diferents mètodes de xifrat. Es veurà exemples on pel simple fet d'haver descodificat un missatge enemic el curs dels esdeveniments va canviar; alguns paradigmes son la batalla de Salamina del segle V aC, l'entrada dels Estats Units a la primera guerra mundial el 1917 o la derrota japonesa de Midway el 1942. En un món on la informació esdevé cada cop més valuosa i la revolució dels mitjans de comunicació ha canviat les relacions entre estats i els ciutadans, la criptologia tindrà un paper cada cop més important en la vida quotidiana.

Cal posar de relleu la dificultat en l'elaboració de la vessant històrica d'aquest treball perquè la criptografia és la ciència del secret i per tant molta informació resta oculta. Molts dels herois que van permetre desxifrar codis enemics i van salvar les seves nacions, no han obtingut el reconeixement públic fins després la seva mort ja que la seva feina mentre durant un bon temps tenia un valor militar i diplomàtic. Les dades de les què disposem es basen en documents desclassificats, un cop han perdut el seu valor estratègic. Hem de ser conscients que hi ha moltes avenços que els estats mantenen en secret, organitzacions com la NSA (National Security Agenct) o GCHQ (Governament

Communications Headquarters) continuen investigant i fent avenços desconeguts al públic.

Una de les tècniques més antigues per ocultar la informació es basa en l'ús de determinades substàncies orgàniques que al assecar-se es tornen invisibles. Per recuperar el missatge ocult tan sols cal acostar el missatge a una font de calor, quan com per art de màgia els caràcters tornaran a ser visibles. Els compostos orgànics, amb un alt contingut de carboni, es carbonitzen sota els efectes de l'escalfor; hi ha moltes substàncies que poden ser utilitzades com a tintes invisibles com el suc de llimona, la llet, el vinagre, la saba d'arbre...Alguns compostos inorgànics també produeixen aquest efecte com a resultat de les reaccions químiques; es pot utilitzar un compost incolor com a tinta invisible i provocar una reacció química que el faci canviar de color i per tant visible a ull nu.

Tot procediment que aconsegueix ocultar físicament el missatge, tapant-lo o fent-lo invisible, tot i que el missatge encara sigui present, rep el nom genèric d'esteganografia - del grec *steganos*, encobert, i *graphein*, escriure -. D'aquestes tècniques de camuflatge el testimoni documentat més antic del que disposem és el de les cròniques d' Herodot. L'esteganografia van tenir un paper fonamental en la història de la civilització occidental: va permetre als Grecs disposar prèviament dels plans de conquesta Perses i conseqüentment poder preparar una defensa.

B.1 L'època clàssica: Xerxes i els grecs

Demaratus, fill d'Ariston, estava exiliat a Persia i li van arribar noticies a Susa, població persa on s'inscriu l'obra d'Ester de Salvador Espriu, de les intencions bel·licistes de l'emperador Xerxes. Va contemplar esfereït el reclutament massiu de tropes i els preparatius de la guerra que es duien a terme a les muntanyes de l'antic regne de l'Elam i va decidir que havia de fer arribar aquesta informació als grecs perquè poguessin preparar la defensa. Com que el risc de ser descobert era molt elevat, va idear un mètode per camuflar la informació i fer que pogués circular sense risc per les carreteres i la frontera de l'imperi persa fins a Atenes i Esparta.

El missatge va ser cisellat en unes taules de fusta i després recobert amb cera calenta. En solidificar-se semblaven taules de fusta aparentment innòcues i van poder així transitar sense que els guardians de les fronteres detectessin res d'estrany. Quan les taules van arribar als seus destinataris, ningú no va veure-hi res de rellevant fins que Cleòmenes, filla de Gorgo, i dona de Leònides va descobrir que sota la cera s'hi amagava un missatge de vital importància cisellat en la fusta.

En el capítol dedicat a les Guerres Mèdiques, Heròdot explica amb detall les argúcies estratègiques mitjançant les quals el general Temístocles, comandant d'una modesta flota de naus gregues va preparar una defensa capaç de guanyar la batalla de la badia de Salamina, infringint així una derrota inesperada a l'aclaparadora força naval de l'imperi persa. Quan Xerxes va atacar el 23 de setembre del 480 a.C, sense el component de la sorpresa, va fer entrar els seus vaixells a la badia amb la princesa persa Artemisia. Els vaixells grecs, que eren més petits i lleugers, haguessin estat destruïts a mar oberta però en la badia tancada, aquest fet es va revelar un poderós aliat. Mentre els vaixells perses xocaven i s'enfonsaven pels cops de vent, els grecs van fer un atac conjunt i van enfonsar la flota enemiga.

Termòpiles anunciava que els soldats perses eren tan nombrosos que quan disparaven enlaire totes les seves fletxes el sol desapareixia. La victòria grega de Salamina, juntament amb la de les Termòpiles i Palatea, va aturar definitivament les aspiracions annexionistes de l'emperador Xerxes envers la lliga de ciutats gregues, encapçalada per Atenes i Esparta; es va mantenir així l'únic oasi on floria la llibertat en tota la Mediterrània oriental.

La història no acaba sense una certa amargor, Gorgo, que podria ser considerada la primera dona criptoanalista, va signar una sentència de mort pel seu marit Leònides, encarregat de la defensa del pas de les Termòpiles. Leònides va morir dirigint un exèrcit d'Espartans que van aturar durant tres valuosíssims dies l'avanç del molt superior en nombre exèrcit persa. Aquesta aferrissada lluita pels ideals ha passat a la història d'Occident com el major exemple de devoció, fe, honor i patriotisme.

B. 2 Edat Mitja i Renaixement: la Reina Maria dels escocesos

La reina Maria d'Escòcia va ser jutjada al castell de Fotheringhay el 15 d'octubre de 1586, després d'anys d'empresonament encara tenia l'aire de dignitat dels membres de la realesa. A la sala plena d'oficials i espectadors, hi havia el tron que simbolitzava l'absència de la reina Elisabet d'Anglaterra, la seva enemiga. Havia estat acusada de planejar l'assassinat de la reina Elisabet per poder apoderar-se del tron d'Anglaterra i la jutjaven per traïció. El fiscal era Sir Francis Walsingham, secretari i conseller de la reina anglesa, que ja havia arrestat altres conspiradors, els havia extret confessions i els havia fet executar.

Walsingham volia provar que la reina Maria era el cervell de la conspiració i que per tant mereixia la mort com els altres, però primer havia de provar la seva participació en el complot. Walsingham sabia que necessitava de guanyar el judici per tres raons si volia poder executar la reina Maria. En primer lloc, molts qüestionaven el dret de la cort anglesa per executar un altre cap d'estat. En segon lloc executant a Maria sense jutjar podria establir un precedent, potser rebels intentarien matar a Elisabet. Per últim, Maria i Elisabet eren cosines, i el vincle familiar feia difícil a Elisabet signar la condemna a mort d'un familiar.

En la conspiració s'hi mesclava un conflicte de caire religiós, els anglesos catòlics volien treure a Elisabet, una protestant, del tron i posar una reina catòlica com Maria. Maria va autoritzar tota la conspiració i el paper de Walsingham era demostrar aquest fet. En els casos de traïció, a l'acusat no li estava permès preparar cap tipus de defensa ni l'ajuda de cap advocat. Maria però confiava que el seu mètode de comunicar-se amb els conspiradors la protegiria: havia xifrat tota la correspondència. En el cas que Walsingham hagués interceptat les cartes no les podia haver llegit i per tant, no podien ser utilitzades com a prova en contra seva. Tanmateix partia de l'assumpció que el xifrat no havia estat trencat.

Walsingham era també el cap de l'espionatge anglès i quan va tenir les cartes xifrades les va entregar a Thomas Phelippes, l'expert en trencaxifrats més brillant d'Anglaterra. La vida de la reina Maria depenia de la seguretat del seu xifrat. Però per entendre els esdeveniments, anirem una mica enrere en la història.

El 24 de novembre de 1542, l'exèrcit d'Enric VIII van eliminar les forces Escoceses a la batalla de Solway Moss; el rei anglès estava a punt de fer-se amb la corona d'Escòcia. Després de la batalla, el rei escocès Jaume V estava tan abatut físicament i mentalment que es va retirar al palau de Falkland: ni tan sols amb el naixement de la seva filla Maria dues setmanes després es va recuperar de la depressió. Una setmana després del naixement de l'heretera del tron, va morir amb tan sols trenta anys.

Maria va ser coronada el 9 de setembre de 1543, amb només nou mesos, reina d'Escòcia a la capella del castell de Stirling. Enric VIII, enlloc d'envair un país governat per una reina infant, que hagués estat mal vist, va decidir intentar casar el seu fill amb Maria. D'aquesta manera podria unir les dues nacions sota un mateix sobirà Tudor.

L'oferta de matrimoni va ser rebutjada per la cort, ja que hi havia una oferta més profitosa, la del dofí de França, Francesc. Es crearia així un sòlid vincle amb la corona francesa, qui protegiria Escòcia davant un atac anglès. Maria es va veure obligada a partir a França el 7 d'agost de 1548 davant la inseguretat del seu país.

Es va casar finalment amb François, a l'edat de setze anys, qui va esdevenir Rei d'Escòcia. Degut a una inflamació de l'orella que es va extendre al cervell, el Rei va morir abans d'anar a ocupar el seu tron a Escòcia i Maria va enviduar. Va tornar a Escòcia el 1561 però es va topar amb una nació transformada, molts escocesos havien passat a l'església protestant i a causa dels seus posteriors matrimonis poc beneficiosos es va anar guanyant enemistats amb els nobles fins que el 1567 va ser empresonada i obligada a abdicar a favor del seu fill Jaume VI.

Quan va escapar de presó es va dirigir cap a Anglaterra perquè esperava que la seva cosina Elisabet l'acollis; tanmateix va cometre un greu error. Va ser empresonada altra vegada ja que significava una amenaça per Elisabet, perquè molts anglesos catòlics la consideraven la veritable reina d'Anglaterra com heretera de Margarida Tudor. Elisabet era una reina il·legítima pels catòlics

perquè era la filla de Anna Bolena, la segona dona de Henry, sense la benedicció papal.

El 1586, després de passar divuit anys a la presó, va escriure una carta al seu fill de dinou anys esperant que l'ajudés a sortir de la presó. Jaume havia estat criat per enemics de Maria i no tenia cap afecte per la seva mare ja que creia que havia estat ella qui havia mort el seu pare. L'odi que sentia per la mare el va dur fins i tot a demanar la mà d'Elisabet tot i que no va haver matrimoni.

El 6 de gener de 1585 va rebre gràcies a Gilbert Gifford, un feix de cartes enviat per partidaris de la seva causa en el continent. Gifford es va convertir en el seu missatger, porta cartes i se'n duia les respostes; tenia una forma força curiosa de fer passar els missatges. Aprofitant els barrils de cervesa que entraven i sortien del castell, amagava furtivament en els taps de suro els missatges que després un dels criats de Maria li lliurava en ma.

La conspiració va començar el març de 1586 quan Babington, el cervell de l'operació, va reunir sis confidents en una taverna. En els propers mesos va emergir un ambiciós pla per allibera la Reina Maria, assassinar la Reina Elisabet i incitar una rebel·lió recolzada per una invasió exterior. El sis de juliol de 1586 Gifford va lliurar a Babington una carta de Maria, en la qual la Reina desitjava conèixer tots els detalls. Legitimaven l'assassinat d'Elisabet amb l'excomunió de la Reina anglesa per part del Papa Pius V el 1570.

Tota la correspondència era xifrada amb un nomenclador, que combinava un codi de substitució monoalfabètica i un conjunt de símbols representant paraules o frases. Gifford era un doble agent, havia ofert els seus serveis el 1585 a Walsingham, secretari de la Reina Elisabet. Walsingham era el cap de seguretat, un personatge maquiavèlic, que havia estès una xarxa d'espionatge a tot el continent: França, Alemanya, Itàlia, Espanya i en altres ciutats com Constantinoble o Trípoli.

Cada cop que Gifford rebia un missatge, el lliurava abans a Walsingam qui en feia una copia i el tornava sense que els conspiradors se'n adonessin. Walsingam que era plenament conscient de la importància de tenir criptoanalistes a la seva disposició, va crear a Londres una escola i va utilitzar Thomas Phelippes com secretari de xifrats. Phelippes era un lingüista que parlava anglès, francès, alemany, italià, castellà i llatí; era un dels millors criptoanalistes de tot Europa.

Quan Phelippes va començar a fer els anàlisis de freqüència de les diverses cartes, era tant sols qüestió de temps que pogués desxifrar els missatges. El més

complicat eren els símbols que representaven paraules però es podien deduir del context. Gradualment va començar a extreure fragments que tenien sentit, i va identificar els punts i les trampes lingüístiques.

Quan Walsingam va rebre les cartes desxifrades per Phelippes que incriminaven a Babington i altres rebels, encara no estava satisfet, ell esperava una resposta de Maria ordenant l'assassinat i firmant així la seva sentència de mort. El 17 de Juliol Maria va respondre a Babington, en la carta quedava palesa el seu coneixement i aprovació de la trama: Walsingam tenia el que necessitava. Tanmateix quedava un darrer pas per destruir la conspiració completament, va ordenar a Phelippes que imités l'escriptura de Maria preguntant per la gent involucrada en la trama, obligant així a Babington a revelar noms.

El xifrat de la Reina Maria mostra que una xifrat amb un codi feble és pitjor que estar sense xifrar perquè tan el receptor com l'emissor confien en la seguretat del canal de comunicació de manera que transmeten informació valuosa sense discreció. L'ús correcte d'un xifrat fort és una gran ajuda per l'emissor i el receptor però l'ús d'un xifrat dèbil pot generar el fals sentiment de seguretat.

El 15 d'Agost de 1586 Babington i sis persones més van ser detingudes i portades a Londres. Les execucions van ser públiques, els que van prendre partit a favor de Maria van ser torturats de manera horrible. El judici de Maria va tenir lloc al castell de Fotheringhay, i va començar el dimecres 15 d'octubre. Molts esperaven veure Maria humiliada, suplicant el perdó però va mantenir la seva dignitat en tot moment. Basava la seva defensa en el desconeixement d'aquell complot, tanmateix davant les proves aportades per Walsingham la seva paraula va quedar en entredit.

El 8 de febrer de 1587 la Reina Maria va ser executada, però per evitar que la seva figura es convertís en la d'un màrtir, Walsingam va cremar tots els seus afers personals. En la última aparició Maria va reafirmar la seva fe cristiana, intentant no mostrar por davant la mort.

Maria i Babington depenien d'un xifrat per mantenir els seus plans en secret, però van viure en una època en la què la criptografia estava molt afeblida pels grans progressos de criptoanàlisi. Tot i que el seu xifrat hagués protegit el missatge d'un simple curiós, no podia mantenir la seguretat davant els atacs continus d'un expert com Phelippes.

B.3 Època contemporània: Telegrama Zimmermann

Des de l'aparició de telègraf les comunicacions van millorar molt, tanmateix van aparèixer altres problemes. Qualsevol persona amb els coneixements necessaris podia col·locar una antena i interceptar les comunicacions, de la mateixa manera que es podia subornar el telegrafista que transmetia els missatges. Per aquest motiu els mètodes de xifrat es van desenvolupar per mantenir els missatges en secret; hem de tenir en compte a principis del segle XX es van començar a establir els primers fils transoceànics que permetien l'intercanvi de missatges a una velocitat desconeguda fins el moment. El control dels mitjans de comunicació va tenir un paper fonamental en la primera guerra mundial ja que obligava l'enemic a transmetre els missatges per mitjans insegurs i per tant els aliats interceptaven moltes de les comunicacions.

Per entendre la rellevància del telegrama de Zimmermann convé situar el context històric de la primera guerra mundial; la guerra que tothom esperava que duraria tan sols unes setmanes durava ja tres llargs anys i no hi havia cap perspectiva d'acabament.

França havia perdut mig milió de vides a la batalla de Verdun i només havia aconseguit resituar la línia de batalla on estava deu mesos abans. Anglaterra havia perdut 60 000 homes a la batalla de Somma en un sol dia i la línia Hindenburg romania intraspassable. Romania, un nou aliat, havia estat ràpidament vençuda i Rússia, el gegant de l'Est havia s'havia retirat el 1917 a causa de la Revolució Russa. La campanya dels submarins alemanys U havia augmentat la pressió econòmica sobre els aliats, i el més sorprenent, tot i la provocació del submarí Lusitania enfonsant un vaixell americà, els Estats Units es mantenien neutrals i amb un president que havia guanyat les eleccions amb l'eslògan: "El que ens va allunyar de la guerra". Aquesta fou la primera guerra submarina de tota la història.

Per als alemanys les coses no pintaven gaire millor, la ofensiva inicial s'havia estancat al Marne i les tropes es mantenien en les trinxeres des de llavors. Els civils s'alimentaven de patates a causa del racionament i nois de quinze i setze anys eren reclutats i enviats als fronts, Grècia i Portugal havien entrat a la guerra pel bàndol aliat i Alemanya no veia la victòria propera.

Alemanya necessitava la rendició dels aliats abans que els EUA entressin a la guerra i mobilitzessin les seves tropes. El Kaiser, Guillem II, va decidir que s'havia de començar el desplegament del submarins U l' 1 de febrer de 1917. En el temps que quedava abans que comencés la batalla submarina, el ministre d'afers

exteriors, Arthur Zimmermann va preparar un pla per mantenir els EUA, un cop haguessin entrat a la guerra, allunyats del camp de batalla continental. Va proposar una aliança amb Mèxic i volia convèncer al seu President per que envaís els territoris de Texas, Nou Mèxic i Arizona, mentre Alemanya enviaria ajuda financera i militar per lluitar contra l'enemic comú. Volia aconseguir que Japó també entrés en guerra; creant així una triple amenaça: els submarins de la costa est, els mexicans del sud i els japonesos de l'oest de manera que els EUA no es poguessin permetre d'enviar tropes als camps de batalla continentals.

Zimmermann va haver d'enviar el telegrama perquè era conscient que els Aliats estaven interceptant les comunicacions transoceàniques. El primer dia de la guerra un vaixell anglès, el Telconia, es va apropar a la costa alemanya i va sabotejar els cables de telègraf, el mitjà de comunicació més segur del que disposaven. A partir de llavors alemanya es va veure obligada a utilitzar emissions de ràdio o cables controlats per altres països. Es van enviar dues copies del mateix missatge, una via Suècia que s'havia mantingut neutral, i l'altra a través del cable americà. Els dos fils tocaven Anglaterra, de manera que el telegrama va ser interceptat pels britànics.

El 17 de gener de 1917, a dos quarts de deu, el reverent William Montgomery, que estava servint com a criptoanalista britànic de la secció diplomàtica en la Sala 40, va entrar a comunicar al almirall Sir William Reginald Hall, director de la intel·ligència naval, que havia estat interceptat el que semblava un missatge important. El criptograma, que ell i el seu col·lega Nigel de Grey havien començat a analitzar, era un llarg missatge d'aproximadament un miler de grups codificats numèricament. Havia estat enviat des de Berlín el 16 de Gener i anava adreçat a l'ambaixador alemany als Estats Units, el compte Johann Heinrich Andreas von Bern-Storff.

El missatge havia estat encriptat utilitzant els codis diplomàtics coneguts com 0075, en els quals els dos criptoanalistes ja havien estat treballant els darrers mesos. El codi 0075 era un codi utilitzat per l'oficina d'Afers Exteriors alemanya; estava format per dues parts, dos zeros i dos dígits que al ser restats donaven com a resultat dos. Era un codi relativament nou distribuït el juliol de 1916 per les missions de Oslo, Bucarest, Copenhaguen, Estocolm, Sofia, Berna, Lugano... però els espies britànics disposaven de suficient material com per intentar trencar el xifratge.

En unes poques hores, De Grey i Montgomery van poder llegir alguns fragments com: "guerra contra els EUA" o "proposar una aliança" que feien palesa la gran importància del missatge enviat per Arthur Zimmermann, el ministre

d'afers exteriors alemany. El missatge contenia les pitjors amenaces però també forçava al president nord-americà Wilson a entrar en guerra del bàndol aliat. Quan de Grey i Montgomery van lliurar el telegrama parcialment desxifrat esperaven que l'almirall Hall passés la informació als americans, portant-los així en guerra contra Alemanya; tanmateix Hall es va limitar a deixar el telegrama a la caixa forta, encoratjant als criptoanalistes a acabar la feina.

L'almirall Hal estava poc disposat a lliurar el parcialment desxifrat telegrama als americans per tres raons: en primer lloc, era conscient que calia acabar de desxifrar-lo per eliminar qualsevol possible malentès; en segon lloc sabia que si donava als americans el telegrama desxifrat de Zimmermann i ells reaccionaven condemnant-lo públicament, llavors els alemanys se n'adonarien que el seu mètode de xifrat havia estat trencat i en crearien per tant un altra tallant així una font d'informació per l'intel·ligència de la Marina britànica.

D'altra banda revelant el missatge, Anglaterra hauria d'admetre que estava interceptant els telegrames de països neutrals com Suècia o EUA: aquest fet hagués molestat i enfadat els EUA. En qualsevol cas, Hall va decidir que no hi havia cap raó per posar en risc una font d'intel·ligència tant valuosa quan els esdeveniments arribarien pel seu propi peu.

L'1 de febrer, com havia ordenat el Kàiser, Alemanya va començar la guerra sense límits. El 2 de febrer, Woodrow Wilson va convocar un gabinet per decidir la resposta americana. El 3 de febrer es va dirigir al Congrés i va anunciar que els EUA continuarien neutrals, actuant com a pacificadors i no com a combatents: això va sorprendre tan a Aliats com als Alemanys.

Hall tot i estar a l'espera dels esdeveniments, no es va quedar amb els braços creuats, només havia aconseguit la informació, ara li faltava posar-la a punt per ser utilitzada pel seu govern. Conseqüentment va concebre un pla que hauria de dissoldre totes les tres dificultats anteriorment mencionades d'un sol cop. Va pensar que el telegrama rebut a Mèxic seria diferent en alguns detalls significatius del que es va enviar des de Berlín; perquè Bernstorff, l'ambaixador als EUA, hauria retirat les ordres dirigides a ell. La data i el nombre de sèrie serien molt probablement diferents.

Si Hall lliurava una copia del telegrama enviat al cònsol de Mèxic, potser els alemanys pensarien que el missatge havia estat robat en territori Americà i no canviarien els codis; es podien crear rumors que confirmessin la historia del robatori. L'habitació 40, el famós laboratori de criptoanàlisi i espionatge, el cònsol a Mèxic no podia desxifrar el codi 0075 i per tant el missatge enviat per Bernstorff s'hauria de rexifrar utilitzant un altra codi que ja coneguessin i això els permetrien acabar d'omplir els forats que els hi restaven. Hall va aconseguir el 5

de febrer, a través de l'agent T, una copia del telegrama enviat a Eckardt xifrat amb el codi 13040.

De Grey i Montgomery havien acabat de desxifrar el missatge; l'almirall Hall, davant la negativa del president americà d'entrar a la guerra, no tenia altra opció que explotar el telegrama Zimmermann. El 23 de febrer de 1917 l'Almirall de la Marina va lliurar al Secretari Britànic d'Afers Exteriors, Walter Page, el telegrama Zimmermann.

> 3retenem començar l'1 de febrer guerra submarina il·limitada. 7ot i així intentarem mantenir els brA neutrals. bn el cas que això nç es produeixi, fem a 0qxic una proposta d'aliança en les bases següents:
>
> cer la guerra junts, fer la pau junts, generçsa ajuda financera, i la cçmprensió des la nçstra part que Mèxic pçt recçnquerir els territçris perduts a Texas, Nçu Mèxic i Arizçna. bls detalls els deixem al seu enteniment.
>
> fnfçrmareu de manera secreta al mresident xde Mèxic] del que ha estat menciçnat prèviament, des del cçmençament de la guerra amb els brA sigui imminent, i afegiu la suggestió que ell hauria, en la seva iniciativa, invitar gapó a adherir-se immediatament i al mateix temps fer de mediadçr entre gapó i nçsaltres.
>
> pi us plau feu veure al mresident que amb la utilització dels nçstres submarins Anglaterra es veurà çbligada a signar la pau.
>
> Zimmermann

Quatre dies més tard, el President Wilson va veure per ell mateix "la prova eloqüent" que Alemanya estava encoratjant l'agressió directa als EUA. El telegrama va ser publicat a la premsa, i finalment, la nació americana es va donar compte de les intencions alemanyes. Els dubtes de si era tot plegat un engany, fabricat per Anglaterra per assegurar l'entrada en la guerra dels EUA, van quedar dissolts quan Zimmermann en una conferència va dir: "no ho puc negar, és cert".

Davant tal revelació els americans es van despertar, la guerra podria estar a la seva frontera i des de la llur fundació no havien mai rebut cap agressió directa en el seu territori. Els texans no es podien creure que Alemanya pretengués donar el seu estat, l'oest es va encendre davant l'amenaça japonesa i el President Wilson, que tres mesos abans es negava a guiar la nació a la guerra, es va dirigir el 2 d'abril al Congrés per demanar ajuda a fer el món una democràcia segura. El telegrama Zimmermann va acabar amb la il·lusió que els

EUA podien dur a terme una política d'unilateralisme, separant-se del que passava al continent europeu.

Ràpidament els americans van aparèixer en els camps de batalla occidentals, aportant aire fresc a les files Aliades. El desxiframent dut a terme per l'habitació 40 va portar els Aliats a la victòria, i als EUA al liderat del món, amb tot el que això ha comportat. Cap criptoanàlisi fins al moment havia comportat tan grans conseqüències, mai abans ni després un sol missatge ha canviat tant el món, durant un temps els trencaxifrats tenien l'historia a la seva mà.