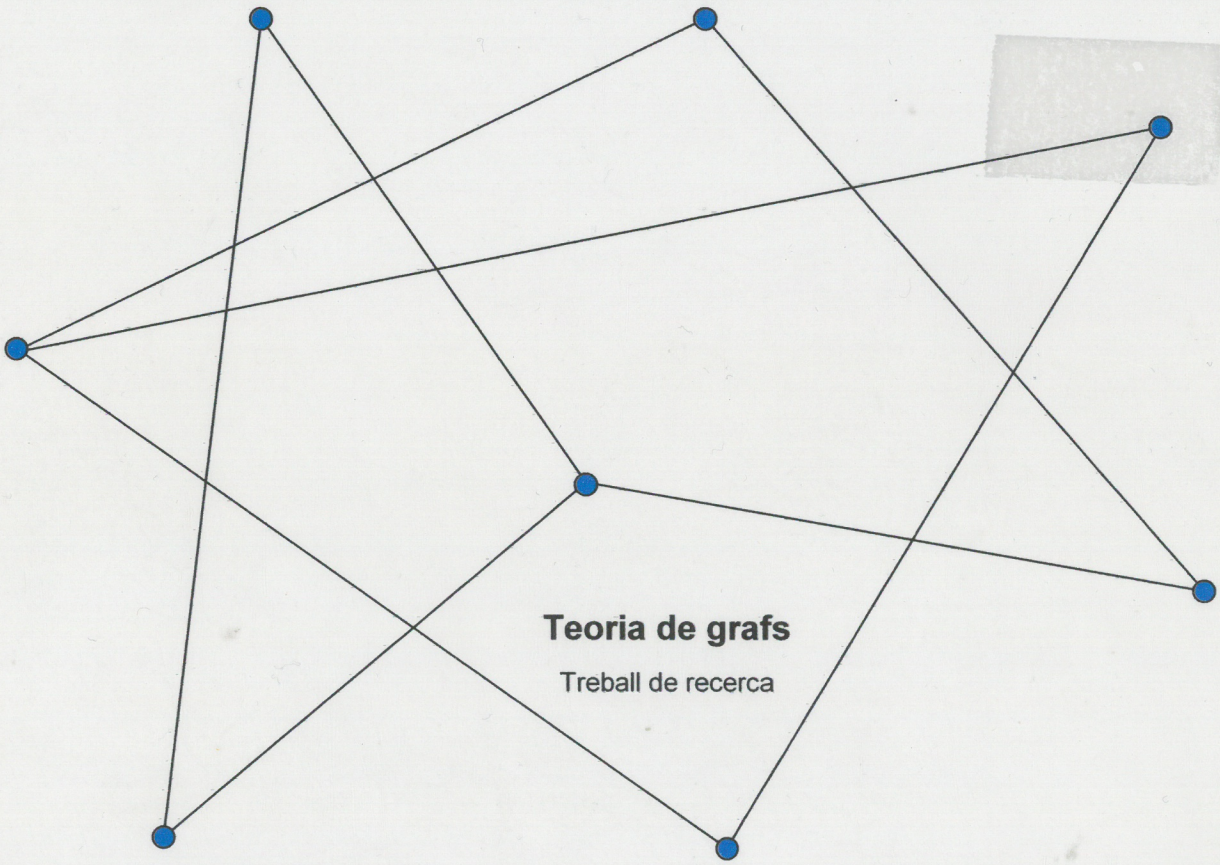
T'has perdut?



3a Edició Premi Poincaré 2006

Tercer premi

T'has perdut?,

Autor:Teresa Pi Torras
Centre: La Salle Bonanova (Barcelona)
Tutor: Sr. Manel Martinez Pascual

per l'anàlisi completa dels algorismes del camí més curt en xarxes i la seva aplicació al mapa de carreteres de Catalunya.

ÍNDEX

1. INTRODUCCIÓ 1

1.1 PRESENTACIÓ DEL TREBALL 1

1.1.1 Criteris d'elecció 1

1.1.2 Objectius 2

1.1.3 Procés 2

1.2 SIMBOLOGIA I TERMINOLOGIA 4

2. TEORIA DE GRAFS 6

2.1 TEORIA 6

2.1.1 Història 6

2.1.2 Conceptes 10

2.1.2.1 Esquema 10

2.1.2.2 Grafs 11

2.1.2.3 Arbres 30

2.1.2.4 Camins i circuits 35

2.1.2.5 Planaritat 43

2.1.2.6 Coloració 45

2.1.3 Problemes 47

2.2 PRÀCTICA 59

2.3 CONCLUSIONS, VALORACIÓ I TREBALL FUTUR 65

2.4 AGRAÏMENTS 68

2.5 BIBLIOGRAFIA I WEBS 68

3. ANNEX 70

3.1 VOCABULARI 70

3.2 CAMPS VINCULATS 72

3.2.1 Algorismes 72

3.2.2 Programació 80

1. Introducció

1.1 Presentació del treball

1.1.1 Criteris d'elecció

Ha estat difícil l'elecció del tema del treball. En primer lloc convenia que el tema complís una sèrie de requisits per adequar-se al que ha de ser un treball de recerca, per exemple: calia que fos quelcom afí als meus interessos i gustos ja que el termini de temps de realització és ampli. Tanmateix amb això no n'hi havia prou. Necessitava que fos alhora un tema relacionat amb alguna de les assignatures que s'imparteixen a l'escola per així poder-me assegurar suport per part del professorat especialitzat en la matèria a l'hora de consultar els dubtes que anirien sorgint. En segon lloc havia de ser un tema que permetés fer un treball amb un cert pes però que, a la vegada, fos prou concret com per evitar un desbordament d'informació que caigués en la superficialitat. Calia que no fos senzill però sí accessible i a l'abast d'una alumna de batxillerat. És a dir, s'havia de trobar l'equilibri en l'amplitud i en la dificultat. Volia que, fos el que fos el que escollís, em permetés fer un petit treball de camp per no limitar-me a elaborar un resum o escrit extret exclusivament de llibres, enciclopèdies o manuals informatius. Calia que inclogués, a més de la teoria, quelcom experimental que em permetés treballar sobre allò prèviament estudiat en l'apartat teòric. I, per últim, original.

Estava interessada en els camps de música, física, lògica i matemàtica. El primer però, presentava en certa manera un problema ja que l'assignatura en qüestió no entra dins el currículum previst en el batxillerat científic –que és el que jo curso –. La física va ser descartada perquè la idea considerada per fer com a treball dins d'aquesta àrea no complia el requisit de l'originalitat i estava vinculada a la música pel que a això s'unia el que hem dit anteriorment. La lògica, així d'entrada, se'm presentava distant i, finalment vaig acabar decidint-me per les matemàtiques.

Ara bé, se'm presentaven dues opcions: fer un treball d'aprofundiment en un tema ja tractat en cursos anteriors o recentment, o bé fer-ne un d'introducció a

un tema desconegut i nou per mi. Després de buscar en llibres de secundària, elaborar llistes amb el temari impartit fins aleshores, cercar pàgines de matemàtiques a internet i interessar-me per les assignatures contingudes en la carrera en qüestió vaig topar-me amb els grafs.

1.1.2 Objectius

- La introducció als grafs.
- L'aplicabilitat en situacions quotidianes dels grafs.
- L'aprenentatge del llenguatge matemàtic.
- La comprensió de textos matemàtics.
- La utilització de diferents programes informàtics.
  - Microsoft Word: pel redactat del treball i la seva presentació.
  - Microsoft Photo Editor: pel tractament de les imatges.
  - Microsoft Power Point: per l'exemplificació del desenvolupament dels algorismes.
  - Paint: pel tractament dels dibuixos.
  - Visual C++: per la part pràctica.

1.1.3 Procés

El treball consta d'una part teòrica i una altra de pràctica. Dins la primera es presenten els conceptes bàsics relacionats amb el tema així com altres aspectes dels grafs.

En primer lloc va ser necessària la recerca d'informació relativa als grafs al ser aquests part d'una matèria completament nova per a mi. Internet, i els llibres: "Grafs: fonaments i algorismes" i "Cuentos con cuentas" van esdevenir les principals eines de consulta.

Vaig elaborar un índex per estructurar el treball en funció dels diferents apartats a tractar.

En segon lloc vaig iniciar la part pràctica. En un principi volia elaborar un graf format a partir de les capitals de comarca de Catalunya com a vèrtexs i les autopistes o carreteres directes entre elles com a arestes utilitzant uns

programes específics que em van ser facilitats pel senyor Benseny (UB). Posteriorment s'aplicaria Dijkstra al graf i s'obtindrien els camins de cost mínim per anar d'un lloc a un altre.

Vaig trobar a www.xtec.es/~aguiu1/imatges/004/p1.htm el "Gran mapa de Catalunya" i el vaig col·locar a Microsoft Photo Editor ja que aquest programa permet conèixer les coordenades dels punts per on passa el ratolí a mesura que aquest es va desplaçant per sobre la imatge. D'aquesta manera vaig determinar les coordenades en píxels dels punts del contorn de Catalunya, així com les dels punts relatius a les capitals de comarca (futurs vèrtexs). Calia aleshores determinar quines serien les arestes del graf. Per fer-ho vaig visitar www.michelin.es/es/home/home.jsp. Mitjançant aquesta pàgina web vaig anar buscant totes les combinacions possibles de carreteres que unissin unes capitals amb les altres i comprovant que existissin. El resultat d'aquesta recerca va ser una llista de 703 arestes. Més tard, utilitzant el Visual Studio 6.0 va ser possible la visualització a través de la pantalla del graf obtingut. Es va veure però, que el nombre d'arestes era tant elevat que no permetia distingir-ne una de la resta amb claredat. Va ser per això que vaig haver de tornar a fer una altra selecció d'arestes, aquest cop més ajustada, seguint el mateix procediment que la vegada anterior però sent més estricta a l'hora de determinar si una aresta passava en el seu recorregut per una capital de comarca que no fos la mateixa que a la que es dirigia. Com a resultes d'aquesta segona tria la llista es va reduir a 150 arestes.

Més tard vaig pensar que seria interessant ampliar la pràctica i, aprofitant que treballant en els camins mínims utilitzava un mapa de Catalunya, vaig decidir d'introduir-hi l'apartat de coloració. La nova idea consistia en transformar el mapa polític de Catalunya en un graf i aplicar-hi una coloració de vèrtexs per tal d'obtenir un mapa pintat de manera que els colors entre comarques que comparteixen una mateixa frontera fossin diferents. El problema va sorgir al veure que el mapa emprat per buscar les coordenades dels punts del contorn de Catalunya no era un mapa polític. Vaig haver doncs, de buscar un altre mapa on apareguessin les fronteres entre comarques i tornar-hi a cercar els punts del contorn a més dels de les fronteres.

I, finalment, els exemples. Quan el treball estava ja bastant avançat, vaig creure necessària la inclusió d'exemples per il·lustrar allò exposat en la teoria i,

sobretot, per mostrar l'aplicació dels algorismes tractats en un graf per facilitar-ne la comprensió ja que la lectura dels algorismes resulta una mica espessa. Els exemples i llurs explicacions s'han elaborat en Microsoft Power Point a causa de les múltiples possibilitats que ofereix.

Tot aquest procés ha estat controlat i supervisat a través de diverses consultes i visites pel tutor del treball.

1.2 Simbologia i terminologia

A continuació es presenta la llista de símbols que apareixen en el treball.

- $\forall$ Per tot
- $\exists$ Existeix
- $\nexists$ No existeix
- $\{\}$ Conjunt
- $\Rightarrow$ Si
- $\Leftrightarrow$ Si i només si
- $\cap$ Intersecció
- $\cup$ Unió
- $\subset$ Inclòs
- $\subseteq$ Inclòs o igual
- $\in$ Pertany
- $\notin$ No pertany
- $\sum$ Sumatori
- = Operador utilitzat per assignar un valor a una variable. Aquesta tindrà el valor de l'expressió que hi haurà a la dreta de =.
- := Operador utilitzat per definir el valor d'una variable. Aquesta en cada moment prendrà el valor de l'expressió que hi haurà a la dreta del :=.
- $\varnothing$ Conjunt buit
- ∞ Infinit

Tot seguit s'exposa un recull de la terminologia emprada en el treball per referir-se als grafs i als seus elements.

- n nombre de vèrtexs
- m nombre d'arestes
- a_i aresta i
- (v_i, v_j) aresta o arc que va de v_i a v_j
- $c(a_i)$ cost de l'aresta ai
- c_{ij} cost de l'aresta (v_i, v_j)
- C_{ij} camí per anar de v_i a v_j
- $c(C_{ij})$ cost del camí que va de v_i a v_j
- () interval on s'exclouen els termes extrems
- [ ] interval on s'inclouen els termes extrems
- w vèrtex inicial
- v* vèrtex actual
- f vèrtex final
- K_n graf complet de n vèrtexs

2. Teoria de grafs

2.1 Teoria

2.1.1 Història

Els inicis de la teoria de grafs se situen a mitjans de s. XVIII amb Euler.

Leonhard Euler va néixer el 15 d'abril de 1707 a Basilea (Suïssa). Fill d'un pare matemàtic, que es va consagrar al clergat, ben aviat va ser introduït en l'art del saber. Ja de petit va mostrar facilitat pels nombres. Estudià a l'Universitat de Basilea, teologia, hebreu i matemàtica amb els germans Bernouilli i aviat va destacar pel seu talent. Al 1727 es traslladà a Sant Petersburg on posteriorment obtingué una càtedra de filosofia natural.

Als 28 anys va solucionar en tant sols tres dies un problema de l'Acadèmia del qual els teòrics deien que es tardaria mesos a trobar solució. Arran d'aquest esforç exhaustiu va perdre la visió d'un ull. No per això es va veure reduïda la seva activitat intel·lectual. Al 1741 es traslladà a Berlín amb la seva dona i els seus tretze fills, per invitació del rei Frederic el Gran, on residí fins al 1766, que tornà a Sant Petersburg. Va morir el 18 de setembre de 1783 a Rússia.

Euler ha estat una figura cabdal en la història de la ciència. Fou un home polifacètic, amb una gran memòria, capacitat de concentració i facilitat pel càlcul mental de quantitats realment grans. Va realitzar estudis sobre astronomia, resolgué problemes de física, medicina, metal·lúrgia i geografia però especialment va destacar en la matemàtica. En aquesta àrea va aportar resultats en la teoria de nombres, el càlcul, l'àlgebra i la geometria de la posició i, es pot dir, que creà noves branques com la topologia combinatòria, el càlcul

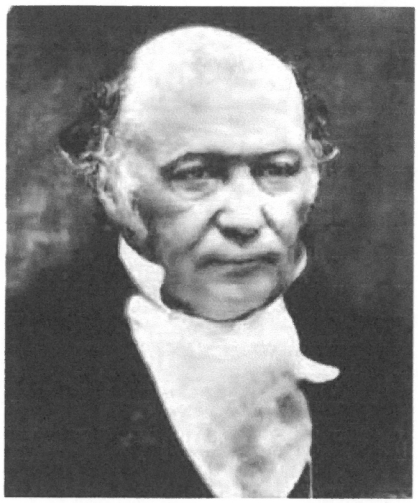
de variacions i la teoria de corbes. El seu llegat en el camp de la matemàtica ha estat immens (va establir les bases de l'anàlisi matemàtic, féu una investigació sobre la curvatura de superfícies, simbolitzà l'arrel quadrada de –1 amb la lletra i, va legalitzar el concepte de nombres complexos, fou el primer en utilitzar el terme de funció, va trobar 59 parelles de nombres amics...) i, prova d'això n'és la quantitat de teoremes (teoremes sobre el residu obtingut per divisió de potències, teorema dels políedres d'Euler...), fórmules (la fórmula d'Euler...), jocs (el quadrat màgic d'Euler...), equacions (l'equació d'Euler-Lagrange, mètode d'Euler per les equacions diferencials...) descobriments (la recta d'Euler...)... que porten el seu nom.

Euler va ser el precursor de la teoria de grafs amb el plantejament i posterior resolució del problema dels ponts de Königsberg i amb la corresponent caracterització del que més tard passarien a anomenar-se circuits i camins eulerians d'un graf. Fou a partir d'aquest treball que nasqué la teoria de grafs, i s'ha anat desenvolupant fins avui en dia amb les aportacions que realitzaren altres matemàtics al llarg de la història.

Al s. XIX Cayley, Hamilton, Kempe i Kirchhoff entre d'altres van contribuir a l'ampliació d'aquesta nova branca de la matemàtica que tot just estava sorgint.

Arthur Cayley va néixer el 16 d'agost de 1821 a Richmond (Anglaterra). Fill de comerciants, va passar els primers anys de la seva vida a Sant Petersburg i, posteriorment va tornar a Anglaterra. Va ingressar al Trinity College l'any 1838 on va realitzar els estudis de matemàtiques i dret. Posteriorment, hi treballà com a professor. Cayley és considerat un dels pares de l'àlgebra linial. Introduí el concepte de matriu i estudià les seves propietats. Va fer destacables contribucions en les branques de teoria de corbes, geometria de determinants i en la teoria dels invariants algebraics. D'altra banda, el seu treball sobre geometria tetradimensional fou la base dels físics del s. XX. En l'àmbit de grafs desenvolupà el concepte d'arbre generador en un graf simètric. Cayley morí un 26 de gener de 1895 a Cambridge.

William Rowan Hamilton nasqué el 4 d'agost de 1805 a Dublín (Irlanda). El seu pare era un home de negocis sense temps i als tres anys el petit William va anar a viure a casa d'un tiet rector, James Hamilton, que fou qui s'encarregà de la seva formació.



Hamilton va ser un nen prodigi: amb tan sols tres anys ja era capaç de llegir, escriure i tenia nocions avançades d'àlgebra; amb cinc ja llegia llatí, grec i hebreu; amb vuit, italià i francès i als catorze àrab, sànscrit i persa. Al 1820 començà a mostrar interès per les matemàtiques i la física. Dos anys més tard, va descobrir una errada de Laplace, llegint la seva obra Mecànica celest.

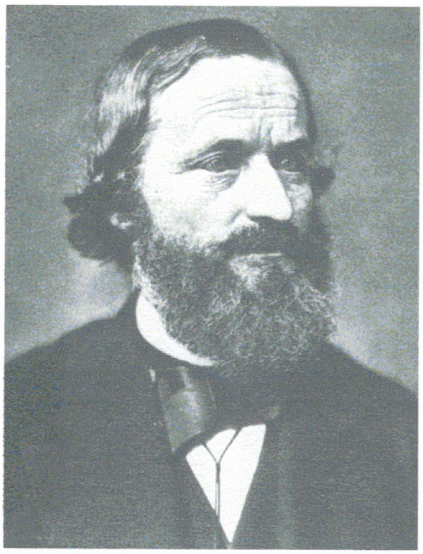
Al 1923 va ingressar al Trinity College on realitzà els seus estudis. De ben jove, amb menys de 22 anys, fou proposat per ocupar les places de professor d'astronomia i d'astrònom reial d'Irlanda. Treballà al mateix Trinity College i a l'observatori de Dunsink. Més tard es va casar amb Maria Bayley i tingué dos fills i una filla. Al 1835 fou nomenat encarregat de la British Assotiation for the Advancement of Science i, posteriorment ocupà el lloc de president de la Royal Irish Academy. Una mala alimentació i una lleugera tendència a l'alcohol van fer que patís de gota en els seus últims anys. Va morir el 2 de setembre de 1865. Hamilton fou un home culte amb interès per les llengües clàssiques i l'observació. Tanmateix, no fou en aquestes disciplines on destacà per la seva tasca i les seves aportacions, sinó en matemàtica i física. Féu un treball sobre òptica per demostrar la naturalesa ondulatòria de la llum, predigué la refracció cònica, establí en dinàmica les funcions que porten el seu nom, intentà unificar òptica i dinàmica i introduí un nou operador diferencial, el ∇, anomenat nabla. També fou el precursor del càlcul vectorial amb la seva descoberta dels quaternions. Hamilton ideà un joc que anomenà Viatge pel món consistent en realitzar un recorregut per 20 ciutats del món corresponents als 20 vèrtexs d'un dodecàedre sense haver de passar més d'un cop per cadascuna d'elles. Fou arran d'això que s'atribueix el nom de hamiltonians als camins i circuits que passen un sol cop per cada vèrtex d'un graf.

Alfred Bray Kempe nasqué el 6 de juliol de 1849 a Kensington, Londres (Anglaterra). Era el tercer fill del rector de Picadilly. Anà a l'escola de St. Paul,

d'alta reputació acadèmica on també rebé formació musical. Kempe aviat destacà per la seva bona veu.

Més tard estudià al Trinity College de Cambridge. Fou deixeble de Cayley i al 1872 es graduà en matemàtiques. No obstant es féu advocat i s'especialitzà en llei eclesiàstica. Al 1877 es casà amb Maria Bowman. Malgrat haver escollit una professió que, en certa manera s'allunyava de les seves aficions, el seu amor per la música i les matemàtiques no minvà i ambdues foren arts a les que dedicà gran part del seu temps. Escrigué alguns articles sobre matemàtiques però se'l coneix, sobretot, pel teorema dels 4 colors. Al 1850, Francis Guthrie, un estudiant anglès es plantejà un problema consistent en trobar el menor nombre de colors possible per pintar un mapa polític de manera que països veïns no rebessin el mateix color. Guthrie ho consultà al seu germà Frederick i aquest ho preguntà a un professor de matemàtiques aficionat als trencaclosques, De Morgan. Augustus De Morgan de seguidà començà a abordar el problema però no va trobar-hi una solució convincent. Envià una carta a Hamilton però Hamilton se'n va mantenir al marge. El problema es va anar estenent fins que Cayley al 1878 el proposà a la London Mathematical Society. Al cap de poc, Kempe proposà una demostració on mostrava que quatre colors eren suficients. Aquesta es va mantenir durant els següents onze anys fins que el matemàtic Heawood trobà un error en la demostració de Kempe. Engrescat amb aquesta troballa dedicà la resta de la seva vida a intentar resoldre el problema. Malgrat que arribà a una sèrie de conclusions (com que són necessaris 7 colors per pintar un mapa sobre la superfície d'un pneumàtic o 6 per fer-ho en la cinta de Möbius) i elaborà una demostració vàlida per a cinc colors, no aconseguí fer-ho per als quatre que havia proposat Kempe. No fou fins al 1950 que Heesch es plantejà la resolució del problema mitjançant un ordinador. Va anar desenvolupant tècniques que posteriorment ajudaren a Appel i Haken demostrar definitivament el teorema dels quatre colors al 1976. Kempe morí un 21 d'abril a Londres.

Kirchhoff va néixer el 12 de març de 1824 a Königsberg (Prússia). Era fill d'un advocat i estudià a la Universitat de Königsberg. Més tard es casà i es traslladà amb la seva dona a Berlín on exercí de professor. Conegué a Bunsen, amb qui realitzaria el seu treball sobre espectroscòpia. Kirchhoff feu importants aportacions en circuits elèctrics, espectroscòpia i emissió de radiació de cos negre. El seu treball sobre la teoria de circuits fou el que contribuí al desenvolupament de la teoria de grafs. Morí el 17 d'octubre de 1887.



Al llarg del s. XX l'interès pels grafs anà augmentant però, sobretot, fou a partir de la segona guerra mundial quan s'hi feren més avenços. Personatges com Berge, Edmonds, Erdös, Harary, Kuratowsky, Lovász, Tutte, Withney, Kruskal, Prim, Dijkstra, Ford, Floyd, Goldberg i Tarjan, entre molts d'altres, van permetre amb les seves corresponents aportacions el desenvolupament de la teoria de grafs fins arribar a avui en dia.

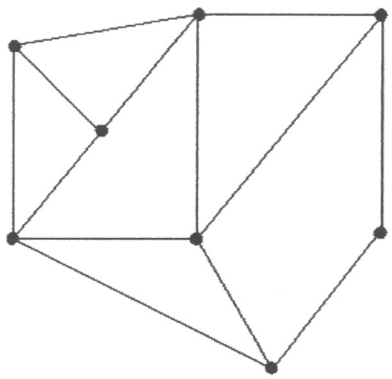
2.1.2 Conceptes

2.1.2.1 Esquema

(Veure esquema.ppt
del CD que s'adjunta)

2.1.2.2 Grafs

Un **graf** és una representació gràfica formada per un conjunt de punts anomenats **vèrtexs** i un conjunt de línies denominades **arestes** o **arcs** tals que cada línia uneix dos dels punts.



Graf G_1

La figura de l'esquerra mostra un graf (G_1).
Els punts negres són els vèrtexs i estan units entre si per les línies.

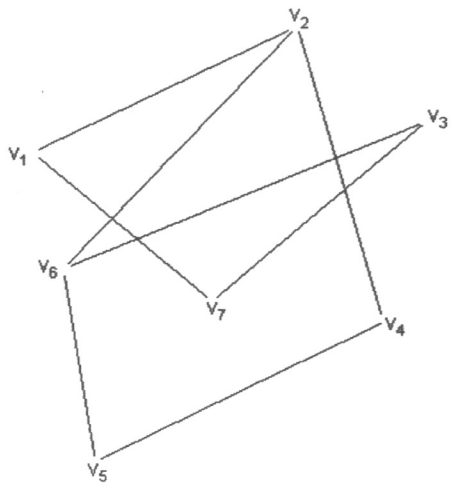
Matemàticament un graf es designa de la següent manera:

$$G(V,A)$$

On V representa el conjunt de vèrtexs i A el conjunt d'arestes:

$$V=\{v_1,v_2, \dots ,v_n\}$$

$$A=\{a_1,a_2, \dots ,a_q\}$$



graf G_2

Aquest graf G_2 té 7 vèrtexs i 8 arestes.

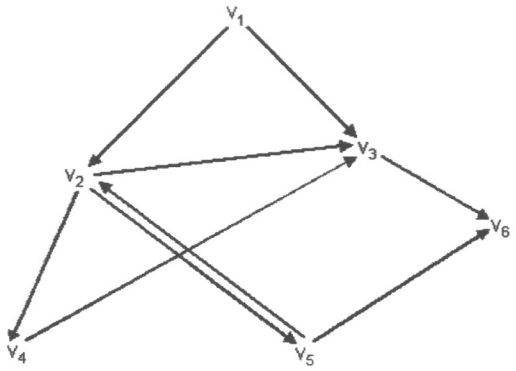
Els vèrtexs de G_1 són:*
$V_2 = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$

I les seves arestes són:
$A_2 = \{(v_1, v_2), (v_1, v_7), (v_2, v_6), (v_2, v_4), (v_3, v_6), (v_3, v_7), (v_4, v_5), (v_5, v_6)\}$

* En segons quins casos no indicarem els vèrtexs amb punts sinó amb l'etiqueta "v_i" per facilitar-ne la referència

Un graf pot ser dirigit, simètric o mixt.

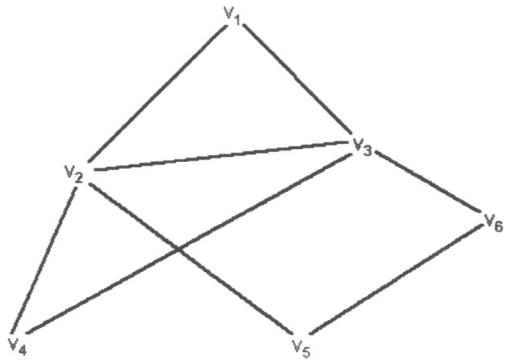
- El **graf dirigit** és aquell les línies del qual tenen una orientació determinada.



Graf G_3

G_3 és un exemple de graf dirigit perquè totes les seves línies estan orientades.

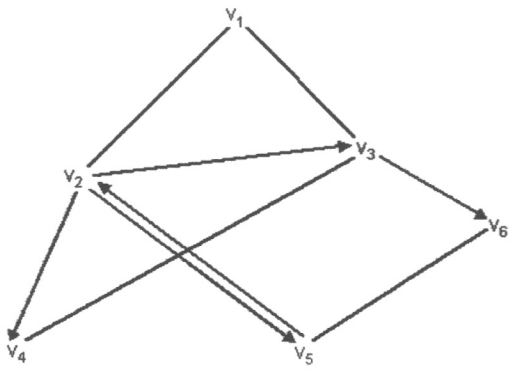
- El **graf simètric** és aquell que no té les línies orientades.



Graf G_4

G_4 és un exemple de graf simètric ja que les seves línies no tenen un sentit determinat.

- El **graf mixt** és aquell que té tant línies orientades com no orientades.



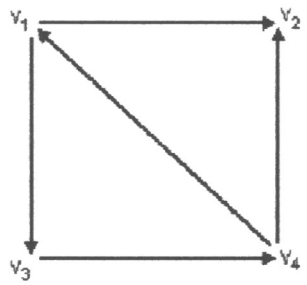
Graf G_5

G_5 és un exemple de graf mixt perquè té línies orientades i d'altres que no ho estan.

▪ VÈRTEXS

Podem diferenciar en un graf dirigit dos tipus de vèrtexs en funció del sentit de l'arc i en relació a un vèrtex v_i:

- Vèrtexs **antecessors** de v_i: són tots els vèrtexs des d'on es pot arribar a v_i. El conjunt format per aquests vèrtexs es designa $\Gamma^{-1}(v_i)$.
- Vèrtexs **successors** de v_i: són tots els vèrtexs on es pot accedir des de v_i. El conjunt format per aquests vèrtexs es designa $\Gamma(v_i)$.

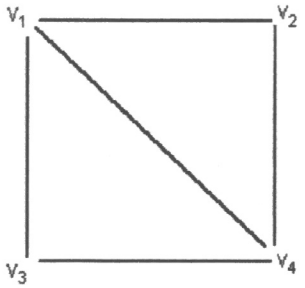


Graf G_6

v_1 és un vèrtex del graf dirigit G_6.
v_4 és l'únic vèrtex antecessor de v_1.
Per tant, $\Gamma^{-1}(v_1) = \{v_4\}$
v_2 i v_3 són els vèrtexs successors de v_1.
Per tant, $\Gamma(v_1) = \{v_2, v_3\}$

En un graf simètric al no poder-los diferenciar parlarem de:

- Vèrtexs **adjacents** o **veïns**: són aquells units per una aresta. El conjunt de vèrtexs adjacents a un v_i es designa $\Gamma(v_i)$.



Graf G_7

v_1 és un vèrtex del graf simètric G_7.
v_2, v_3, v_4 són els vèrtexs adjacents a v_1.
Per tant, $\Gamma(v_1) = \{v_2, v_3, v_4\}$

Grau d'un vèrtex

Tenint en compte l'adjacència d'un vèrtex entra en joc el concepte de grau d'aquest vèrtex.

En un graf dirigit podem parlar de:

- Grau exterior de v_i: el nombre de vèrtexs on es pot anar des de v_i. És a dir:

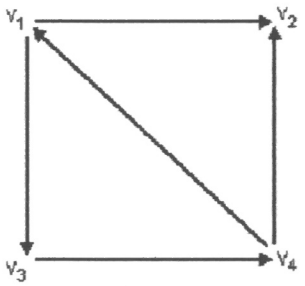
$$|\Gamma(v_i)|$$

- Grau interior de v_i: el nombre de vèrtexs des dels quals es pot arribar a v_i. És a dir:

$$|\Gamma^{-1}(v_i)|$$

- Grau de v_i: la suma dels graus interior i exterior de v_i. És a dir:

$$|\Gamma(v_i)| + |\Gamma^{-1}(v_i)|$$



Graf G_6

v_1 és un vèrtex del graf dirigit G_6.
v_1 té <u>dos</u> vèrtexs successors: v_2 i v_3.
Per tant, el grau exterior de v_1 és 2.

$$/\Gamma(v_1)/ = 2$$

v_1 només té <u>un</u> vèrtex antecessor: v_4
Per tant, el grau interior de v_1 és 1.

$$/\Gamma^{-1}(v_1)/ = 1$$

Així doncs, el grau de v_1 és 3.

$$/\Gamma(v_1)/ + /\Gamma^{-1}(v_1)/ = 2 + 1 = 3$$

En un graf simètric es parla tant sols de:

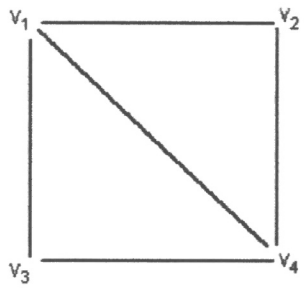
- Grau de v_i: el nombre de vèrtexs adjacents a v_i.

$$|\Gamma(v_i)|$$

En els grafs simètrics tenen una especial importància els valors màxim i mínim per als graus dels vèrtexs de G. Es defineix:

$$\Delta(G)= \max_{v_i \in V} \{\text{grau}(v_i)\}$$

$$\delta(G)= \min_{v_i \in V} \{\text{grau}(v_i)\}$$



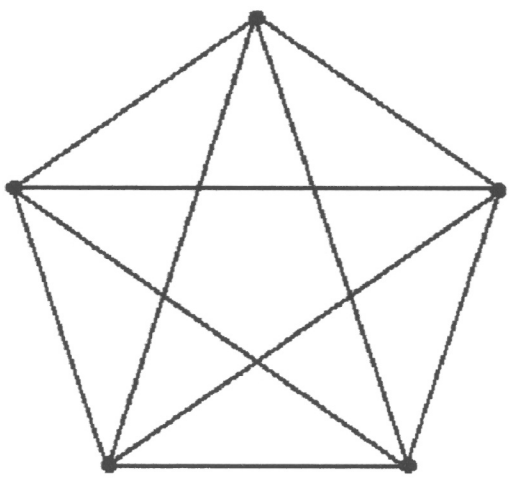
Graf G_7

v_1 és un vèrtex del graf simètric G_7.
v_1 té tres vèrtexs adjacents: v_2, v_3 i v_4.
Per tant, el grau de v_1 és 3.

$$|\Gamma(v_1)| = 3$$

$$\left.\begin{array}{l} |\Gamma(v_1)| = 3 \\ |\Gamma(v_2)| = 2 \\ |\Gamma(v_3)| = 2 \\ |\Gamma(v_4)| = 3 \end{array}\right\} \quad \begin{array}{l} \Delta(G_7) = 3 \\ \delta(G_7) = 2 \end{array}$$

Un graf G simètric on tots els vèrtexs siguin de grau *k* s'anomena *k-regular* .



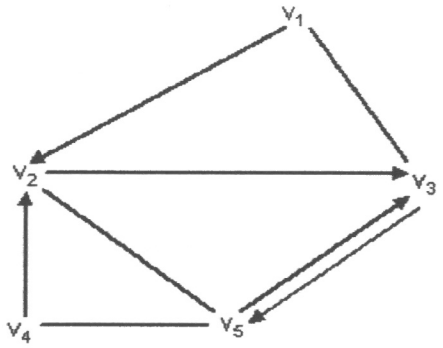
Graf G_8

Tots els vèrtexs de G_8 són de grau 4.
Per tant G_8 és 4-regular.

En un graf mixt parlem de:

- Grau de v_i: la suma del nombre d'arestes i arcs que hi porten i/o en surten. És a dir:

$$|\Gamma(v_i)_{arcs}| + |\Gamma^{-1}(v_i)_{arcs}| + |\Gamma(v_i)_{arestes}|$$



Graf G_9

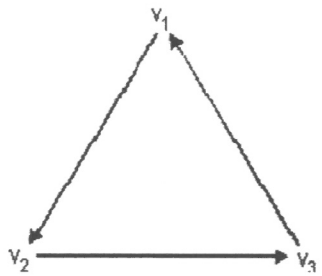
v_2 és un vèrtex del graf mixt G_9.
Arriben dos arcs a v_2.
Surt un arc de v_2.
v_2 també forma part d'una aresta: (v_2, v_5).
Per tant, el grau de v_2 és 4.

$$|\Gamma(v_2)_{arcs}| + |\Gamma^{-1}(v_2)_{arcs}| + |\Gamma(v_2)_{arestes}| = 1 + 2 + 1 = 4$$

▪ LÍNIES

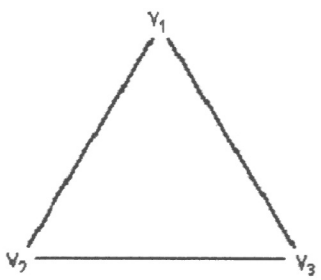
Les línies rebran els noms de:

- **Arcs**: si estan orientades. Apareixeran només en grafs dirigits o mixtos.
- **Arestes**: si no tenen orientació. Apareixeran només en grafs simètrics o mixtos.
- **Llaços**: quan surti del mateix vèrtex on es dirigeixi. Pot aparèixer tant en grafs dirigits, com mixtos, com simètrics.



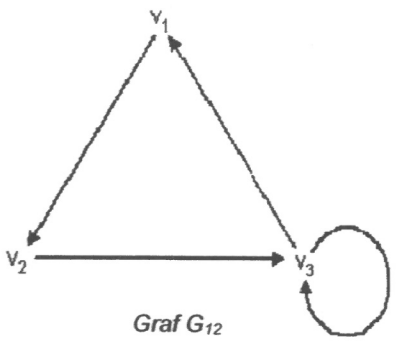
Graf G_{10}

G_{10} és un graf dirigit, les seves línies s'anomenen arcs.



Graf G_{11}

G_{11} és un graf simètric, les seves línies s'anomenen arestes.

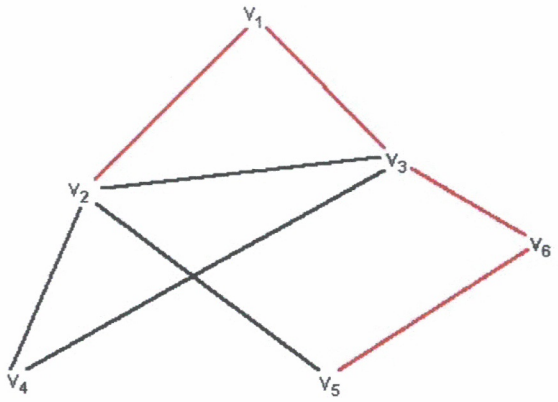


Graf G_{12}

G_{12} és un graf dirigit (té arcs) i té un llaç en el vèrtex v_3.

Seqüències d'arestes/arcs

Un **camí** és una seqüència ordenada d'arestes o arcs d'un graf G tal que el vèrtex final d'una aresta (arc) sigui el vèrtex inicial de l'aresta (arc) següent.



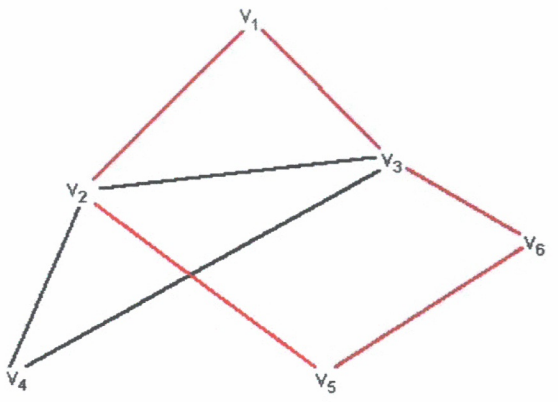
Graf G_4

La figura de l'esquerra mostra un camí per anar de v_2 a v_5.
Aquest camí està format per la seqüència d'arestes (v_2, v_1), (v_1, v_3), (v_3, v_6), (v_6, v_5).

Podem distingir entre:

- Camí simple: aquell que no repeteix arestes (arcs). (Veure camins eulerians)
- Camí elemental: aquell que no repeteix vèrtexs. (Veure camins hamiltonians)

Un **circuit** és un camí tancat.

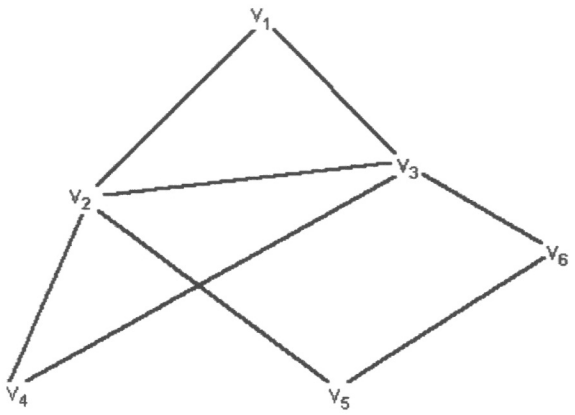


Graf G_4

La figura de l'esquerra mostra el circuit format per les arestes (v_2, v_1), (v_1, v_3), (v_3, v_6), (v_6, v_5), (v_5, v_2).

La **distància** entre dos vèrtexs v_i i v_j és el nombre d'arestes contingudes en el camí més curt existent entre v_i i v_j. Es designa:

$$d(v_i, v_j)$$



Graf G_4

En G_4 la distància entre v_1 i v_5 és 2.

$$d(v_1, v_5) = 2$$

Aquesta ve donada pel nombre d'arestes que formen el camí més curt que en aquest cas és (v_1, v_2), (v_2, v_5).

▪ TIPUS DE GRAFS I ELS SEUS ELEMENTS

Tipus de grafs

Atenent/atenyent a una sèrie de característiques presentades pels grafs anomenem:

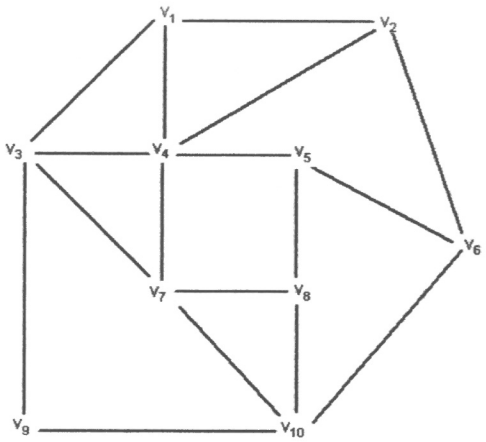
- **Graf trivial**: aquell graf que consta d'un sol vèrtex i cap aresta.

Graf G_{13}

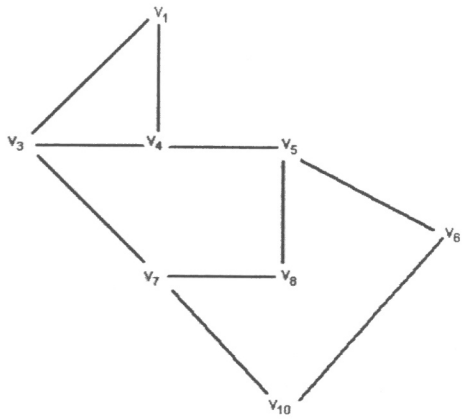
G_{13} és un graf trivial.

- **Graf parcial** de G(V,A): aquell graf les arestes i vèrtexs del qual estan continguts en G(V,A). És a dir:

$$G_P(V_P,A_P) \Rightarrow V_P \subseteq V \text{ i } A_P \subseteq A$$



Graf G_{14}

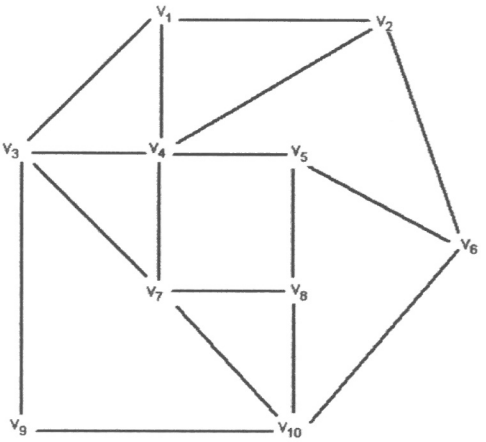


*Graf parcial de G_{14}**

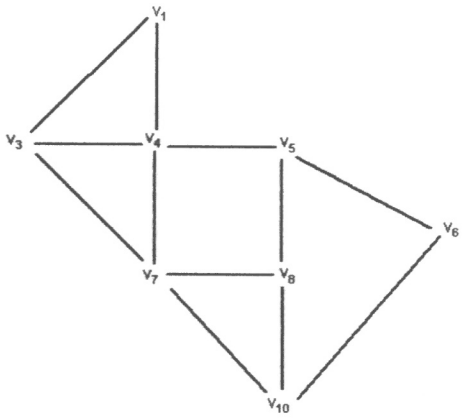
* Els nombres dels vèrtexs es corresponen amb els dels vèrtexs de G_{14}.

- **Subgraf** de G(V,A): graf obtingut a partir d'una sèrie de vèrtexs de G i les arestes que els uneixen. És a dir:

$$G_S(V_S,A_S) \Rightarrow V_S \subseteq V \text{ i } \forall v_i \subseteq V_S \Rightarrow \Gamma_S(v_i) = \Gamma(v_i) \cap V_S$$



Graf G_{14}



Subgraf de G_{14}

Els nombres dels vèrtexs es corresponen amb els dels vèrtexs de G_{14}.

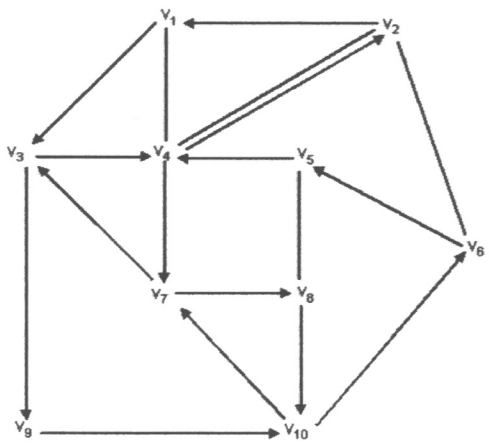
- **Subgraf maximal** de G respecte a *p* (éssent *p* una propietat aplicable a G): subgraf de G que compleix *p*, tal que no existeix cap altre subgraf de G que contingui els vèrtexs del subgraf maximal de G complint *p*. És a dir:

  $\hat{G}_S$ és subgraf maximal de G $\Rightarrow$ verifica *p* i $\nexists$ G_S tal que $\hat{V}_S \subset V_S$ i verifiqui *p*

- **Subgraf minimal** de G respecte a *p*: subgraf de G que compleix *p*, tal que no existeix cap altre subgraf de G els vèrtexs del qual estiguin continguts en el subgraf minimal de G que també compleixi *p*. És a dir:

  $\hat{G}_S$ és subgraf minimal de G $\Rightarrow$ verifica *p* i $\nexists$ G_S tal que $V_S \subset \hat{V}_S$ i verifiqui *p*

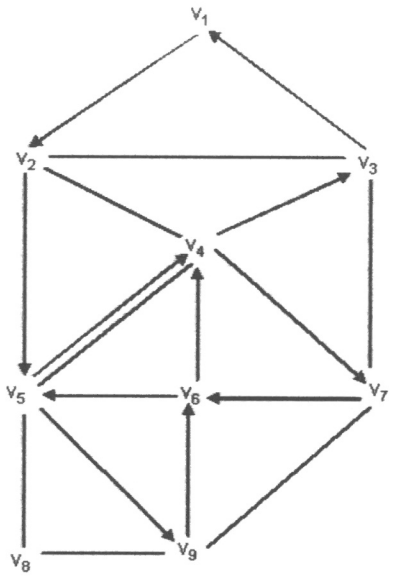
- **Graf mixt balancejat**: graf mixt cada vèrtex del qual té el mateix nombre d'arcs que en surten que d'arcs que hi porten.



Graf G_{15}

G_{15} és un graf mixt balancejat perquè conté arcs i arestes i perquè en cada vèrtex hi arriba el mateix nombre d'arcs que en surt.

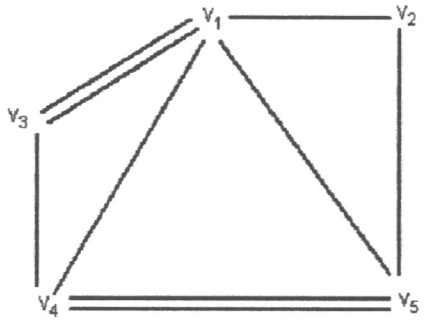
- **Graf mixt parell**: graf mixt cada vèrtex del qual és de grau parell.



Graf G 16

G_{16} és un graf mixt parell perquè tots els seus vèrtexs són de grau parell.
(G_{16} és també un graf mixt balancejat).

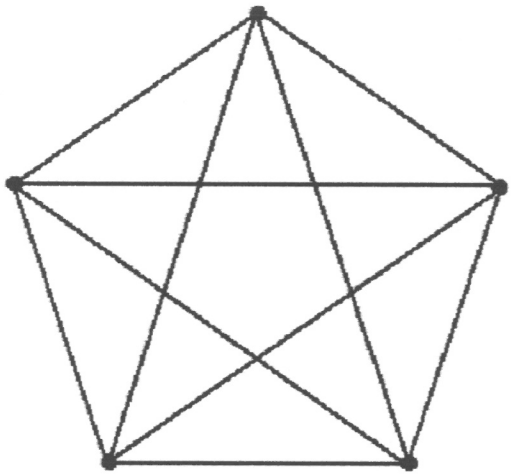
- **Multigraf**: aquell graf que té arcs/arestes repetits/des.



Graf G_{17}

G_{17} és un multigraf perquè les arestes (v_1, v_3) i (v_4, v_5) estan duplicades.

- **Graf complet**: aquell graf sense llaços que conté el màxim nombre d'arestes/arcs possible.
    - El graf simètric i complet de t vèrtexs es designa K_t.



Graf G_8

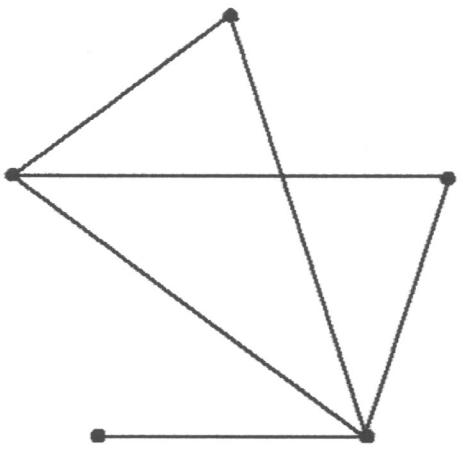
La figura de l'esquerra és un graf complet perquè cada vèrtex està unit mitjançant una aresta a la resta de vèrtexs.
Com que G_8 és simètric, té 5 vèrtexs i és un graf complet es pot designar també K_5.

- **Graf complementari** d'un altre graf no complet: aquell graf que afegit al graf al qual complementa dona un graf complet. Es designa:

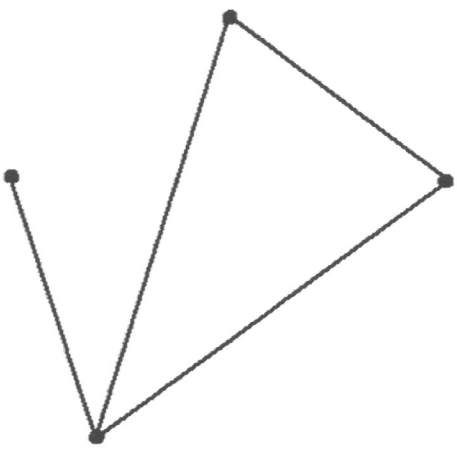
$$\overline{G}(V, \overline{A})$$

Dit d'altra manera: "El graf complementari $\overline{G}(V, \overline{A})$ corresponent a un graf G(V,A) és el graf format per totes les arestes que li manquen a G per a ésser complet"*.

**Grafs: fonaments i algorismes*

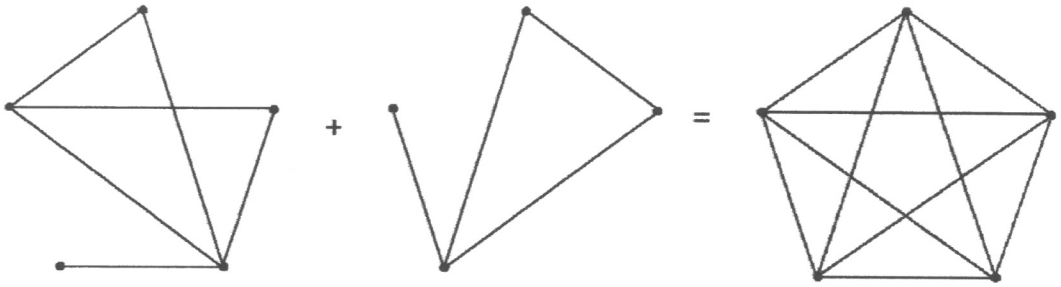


Graf G_{18}



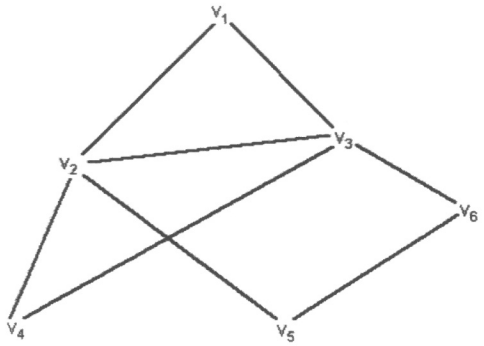
Graf $\overline{G}_{18}$

El graf $\bar{G}_{18}$ és complementari de G_{18} perquè:



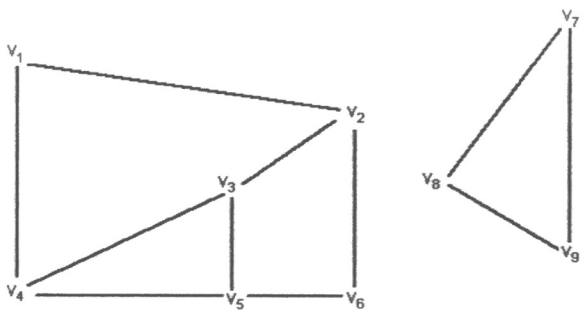
Graf G_{18} *Graf $\bar{G}_{18}$* *K_5*

- **Graf simètric connex**: graf simètric on cada parella de vèrtexs està conectada almenys per un camí.



Graf G_4

G_4 és un graf simètric connex.



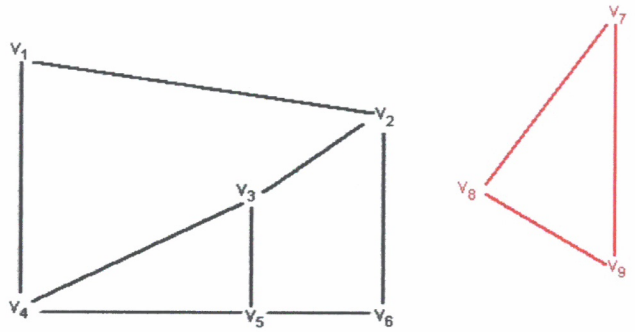
Graf G_{19}

G_{19}, en canvi, és un graf simètric disconnex perquè entre $\{v_1, v_2, v_3, v_4, v_5, v_6\}$ i $\{v_7, v_8, v_9\}$ no existeix cap camí.

Elements dels grafs

Podem trobar en els grafs:

- **Component**: subgraf maximal connex.



Graf G_{19}

G_{19} és un graf simètric disconnex.
El subgraf maximal de G_{19}, $\hat{G}_{S19}(\hat{V}_{S19}, \hat{A}_{S19})$ format per:

$\hat{V}_{S19} = \{v_7, v_8, v_9\}$

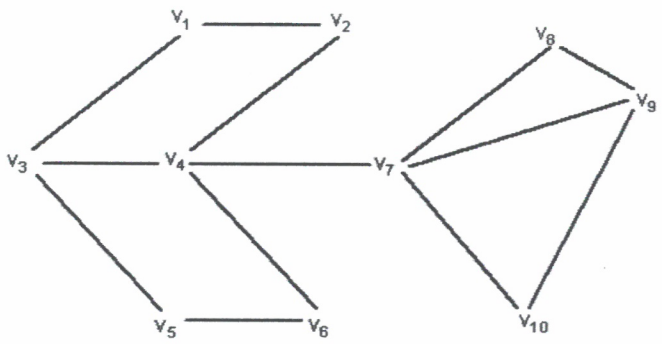
$\hat{A}_{S19} = \{(v_7, v_8), (v_7, v_9), (v_8, v_9)\}$

és un component de G_{19}.
G_{19} és un graf amb 2 components.

- **Pont**: aresta a_j en G tal que G sense a_j té més components que amb ella.

Si $a_j \subseteq A$;

a_j és pont a G $\rightarrow$ |components de G(V,A)| < |components de G(V,A−{a_j})|



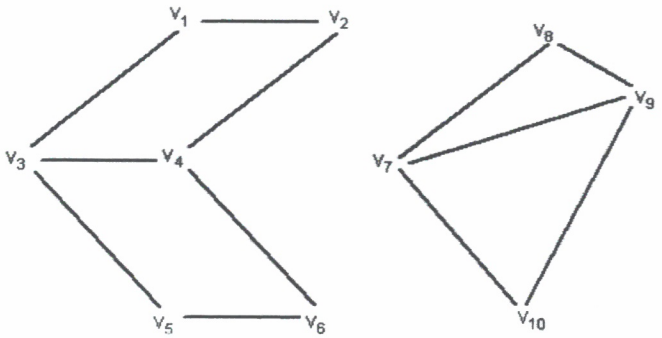
Graf G_{20}

G_{20} és un graf simètric d'un sol component.
(v_4, v_7) és una aresta de G_{20}.

Aquesta aresta és un pont de G_{20} perquè si s'elimina el graf resultant tindrà més components que G_{20}, en tindrà dos.

Aquest graf serà el resultat de fer:

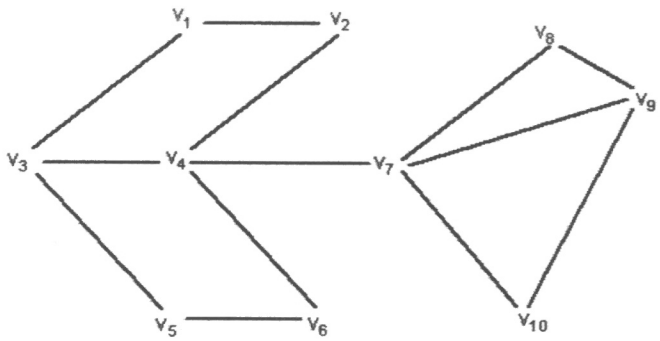
$G_{20} - (v_4, v_7)$



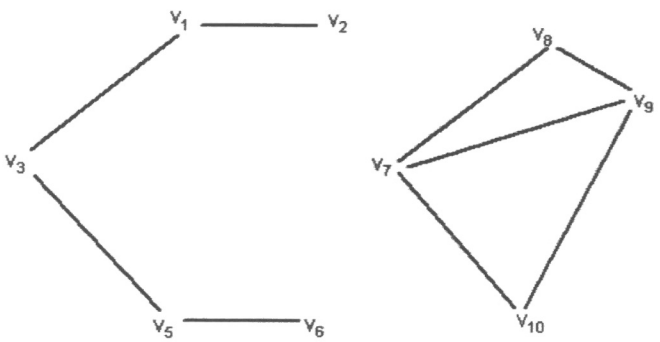
- **Articulació**: vèrtex v_i en G tal que G sense v_i ni les arestes que tenen v_i com a vèrtex final té més components que amb v_i i les arestes que l'uneixen als seus vèrtexs adjacents.

Si $v_i \subseteq V$ i $A_i \subseteq A$, essent A_i={arestes amb v_i com a vèrtex terminal}

v_i és articulació $\rightarrow |\text{components de } G(V,A)| < |\text{components de } G(V-\{v_i\}, A-A_i)|$



Graf G_{20}



G_{20} és un graf simètric d'un sol component.
v_4 és un vèrtex de G_{20}.

v_4 és una articulació de G_{20} perquè si l'eliminem i, amb ell, les arestes que l'uneixen a la resta de vèrtexs de G_{20}, el graf resultant tindrà més components que G_{20}, en tindrà dos.

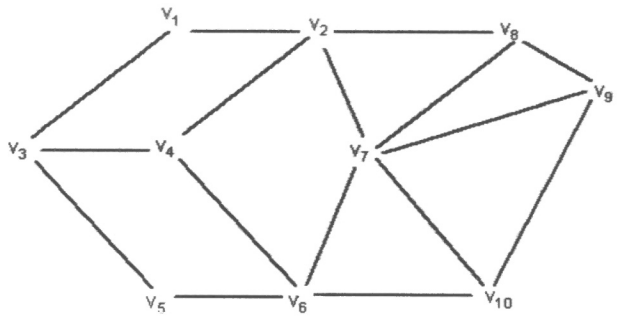
Aquest graf serà el resultat de fer:

$G_{20} - v_4$

- **Separador**: conjunt d'arestes o vèrtexs en G tal que G sense ells/es té més components que amb.

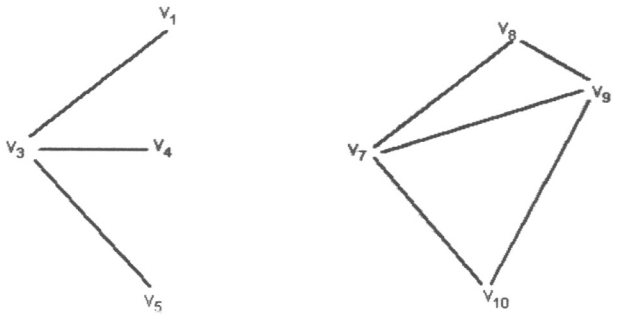
Si U:=conjunt de vèrtexs o arestes de G

U és separador→ | components de G | < | components de G–U |



Graf G_{21}

G_{21} és un graf simètric d'una sola component.
Els vèrtexs v_2 i v_6 formen part de G_{21}.

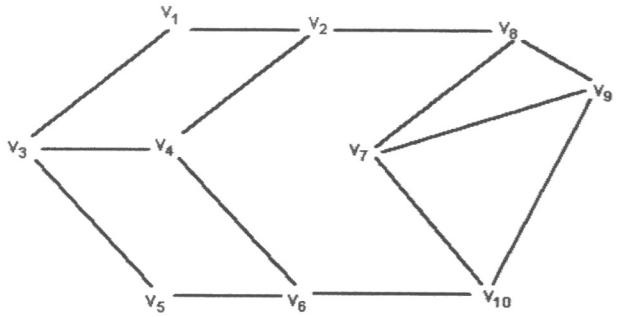


v_2 i v_6 juntament amb les arestes que els tenen en un dels seus extrems com (v_1, v_2), (v_4, v_2), (v_7, v_2), (v_8, v_2), (v_5, v_6), (v_4, v_6), (v_7, v_6), (v_{10}, v_6) constitueixen un separador de G_{21} ja que sense ells el graf resultant té dues components

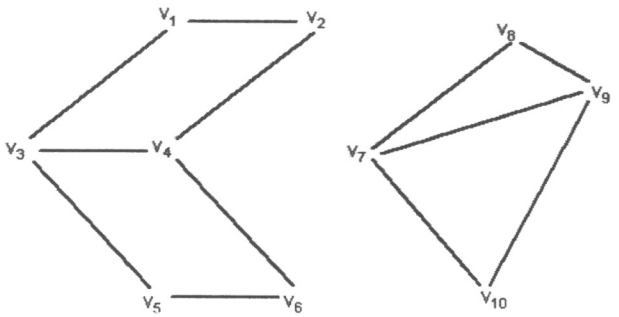
- **Tall**: conjunt minimal d'arestes en un graf G simètric i connex sense les quals G resulta disconnex.

Si A'⊂A

A' és tall →G(V,A–A') és disconnex



Graf G_{22}



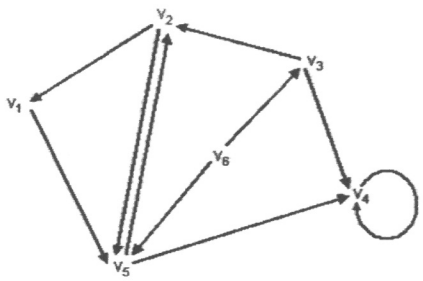
El graf G_{22} és un graf simètric connex.
Les arestes (v_2, v_8) i (v_6, v_{10}) són arestes de G_{22}.

El conjunt format per aquestes dues arestes és un tall ja que és el conjunt minimal d'arestes necessari per tal que la seva supressió de G_{22} origini un nou graf disconnex.

▪ EMMAGATZEMATGE DE GRAFS

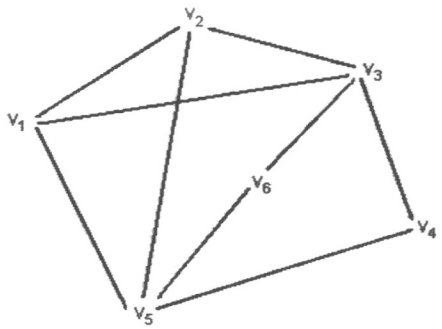
Hi ha dos sistemes d'emmagatzematge de grafs: mitjançant les matrius d'adjacència o les llistes d'adjacència.

- Matriu d'adjacència: Aquesta indica la relació d'adjacència entre cada parella de vèrtexs del graf en qüestió. És a dir, el nombre que apareix en la *i*-èsima fila i en la *j*-èsima columna determina el nombre d'arestes o arcs que van des d'*i* fins a *j*.
  La matriu d'adjacència d'un graf simètric serà també simètrica i per tant només caldrà fixar-se en la informació d'un dels dos costats de la diagonal principal.



Graf G_{23}

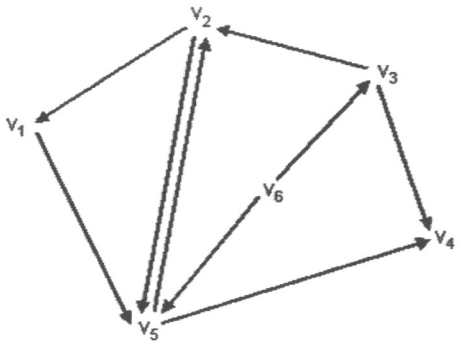
$$A(G_{23}) = \begin{pmatrix} 000010 \\ 100010 \\ 010100 \\ 000100 \\ 010100 \\ 001010 \end{pmatrix}$$



Graf G_{24}

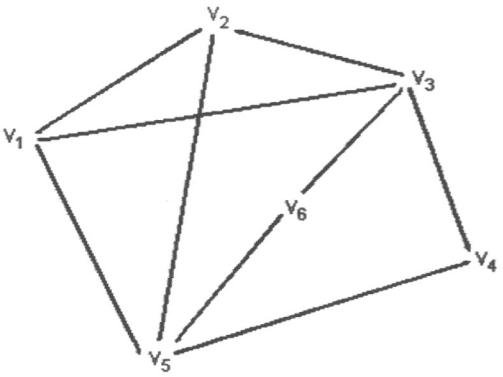
$$A(G_{24}) = \begin{pmatrix} 011010 \\ 101010 \\ 110101 \\ 001010 \\ 110101 \\ 001010 \end{pmatrix}$$

- Llista d'adjacència: Aquest sistema consisteix en elaborar per a cada vèrtex de G una llista on apareguin els seus adjacents en el cas d'un graf simètric o els seus successors en el cas d'un graf dirigit.



Graf G_{23}'

v_1 →	5	/		
v_2 →	1	5	/	
v_3 →	2	4	/	
v_4 →	/			
v_5 →	2	4	/	
v_6 →	3	5	/	



Graf G_{24}

v_1 →	2	3	5	/	
v_2 →	1	3	5	/	
v_3 →	1	2	4	6	/
v_4 →	3	5	/		
v_5 →	1	2	4	6	/
v_6 →	3	5	/		

2.1.2.3 Arbres

Un arbre és un graf connex sense circuits.

La caracterització dels arbres es pot fer a partir de diversos punts de vista recollits en els teoremes següents:

- Teorema: G(V, A) és un arbre ⇔ no conté circuits i n = m+1, essent n el nombre de vèrtexs i m el nombre d'arestes.
- Teorema: G(V, A) és un arbre ⇔ és connex i n = m+1
- Teorema: G(V,A) és un arbre ⇔ conté un sol camí entre cada parella de vèrtexs v_i, v_j.

Arbres generadors

Un arbre generador (spanning tree) d'un graf G(V, A) és un graf $G_P(V_P, A_P)$ tal que G_P sigui un arbre i $V_P = V$.

Un graf pot tenir més d'un arbre generador. Per esbrinar-ne el nombre caldrà procedir de la següent manera:

- En el cas d'un graf dirigit G(V, A):

  El nombre d'arbres generadors amb arrel v_r vindrà donat pel valor del determinant de la matriu D' obtinguda com a resultat de suprimir les r-èsimes fila i columna de la matriu de graus-entrant D.

  La matriu de graus-entrant D = d(ij) és una matriu quadrada tal que:

$$d(ij) = \begin{cases} \text{grau interior } (v_i) & \text{si } i = j \\ -\text{ grau exterior } (v_i) & \text{si } i \neq j \end{cases}$$

- En el cas d'un graf complet K_n:

  El graf K_n generarà n^{n-2} arbres.

Tipus d'arbre

Hi ha diferents classes d'arbres generadors segons la seva utilitat, segons el nombre de vèrtexs que aquests tinguin...

1. Segons el recorregut:

Els arbres BFS i DFS s'utilitzen per conèixer l'estructura interna d'un graf, per resoldre-hi un problema...

arbre	BFS	DFS
nom complet	Breadth first search	depth first search
exploració	Per amplada	Per fondària
explicació	S'obté a partir d'una exploració des d'un vèrtex v_1 a $\Gamma(v_1)$. Es continua l'exploració des del primer vèrtex $v_i \in \Gamma(v_1)$ a $\Gamma(v_i)$. Després des del segon vèrtex $v_j \in \Gamma(v_1)$ a $\Gamma(v_j)$... i així successivament fins esgotar tots els vèrtexs de $\Gamma(v_1)$. A continuació es va repetint el mateix procediment al següent nivell de vèrtexs. S'acaba quan s'ha passat per tots els vèrtexs $v \in V$.	S'inicia l'exploració des d'un vèrtex v_1. Des de v_1 es va a un $v_i \in \Gamma(v_1)$, a continuació es va de v_i a un $v_j \in \Gamma(v_i)$, de v_j a un $v_k \in \Gamma(v_j)$ i així successivament fins arribar a un vèrtex v_r tal que $\Gamma(v_r) = \varnothing$. Al arribar a aquest punt caldrà retrocedir fins a v_{r-1} i buscar un vèrtex $v_t \in \Gamma(v_{r-1})$ per continuar fent l'exploració a partir de v_t. S'acaba quan s'ha passat per tots els vèrtexs $v \in V$.
algorisme a utilitzar	BFS	DFS

2. Segons el cost de les arestes o arcs

En altres casos pot interessar conèixer una solució òptima. Aleshores caldrà buscar arbres de cost mínim en grafs amb costos en les arestes o arcs. En funció del nombre de vèrtexs $z \in Z$ (essent $Z \subseteq V$) que es requereixin en l'arbre a buscar es presentaran tres situacions:

tipus d'arbre	segons $\|Z\|$	segons el graf	segons el cost	algorisme a usar
arbre òptim	$\|Z\| = \|V\|$	G simètric	$c(ij) \geq 0$	Kruskal
				Prim
		G dirigit		Edmonds
	$2 < \|Z\| < n$	graf de Steiner*		Sullivan
	$\|Z\| = 2$**	G simètric	$c(a_i) = c(a_j)$	Moore
		G dirigit	$c(ij) \geq 0$	Dijkstra
			c(ij) no restringit ***	Ford
			c(ij) no restringit	Floyd

* cas particular

** de fet, quan $|Z| = 2$ s'està parlant de camins

** sense circuits de cost negatiu

2.1 $|Z| = |V|$

G(V, A) és un graf simètric amb costos no negatius en les arestes. Per buscar un arbre generador de G que contingui tots els vèrtexs de G caldrà aplicar els algorismes de Kruskal o Prim.

Si G(V, A) en comptes de ser simètric és dirigit aleshores caldrà multiplicar el cost de cada aresta per –1 i aplicar a G l'algorisme d'Edmonds.

2.2 $2 < |Z| < n$

Quan l'arbre generador que es vol buscar conté $|Z|$ vèrtexs – o $|Z| + |U|$, tal que $U \subseteq (V - Z)$ –, essent $Z \subseteq V$ i $|Z| \in (2, n)$ serà un arbre de Steiner per a Z. L'arbre de Steiner per a Z és l'arbre òptim que permet unir els vèrtexs de Z. Depenent de cada cas aquest arbre estarà format solament per Z vèrtexs o per aquests mateixos vèrtexs Z a més a més d'altres vèrtexs U necessaris per tal que el resultat sigui òptim.

Per facilitar la resolució del problema es pot reduir el volum de dades a tractar en l'algorisme d'acord amb els següents punts:

- Si $v_i \in G$, $v_i \notin Z$ i $grau(v_i) = 1 \Rightarrow v_i$ i (v_i, v_j) es podran eliminar de G.
- Si $v_i \in G$, $v_i \notin Z$ i $grau(v_i) = 2 \Rightarrow v_i$ s'elimina i (v_j, v_i) i (v_i, v_k) se substitueixen per una sola aresta (v_j, v_k) de $c_{jk} = c_{ji} + c_{ik}$. Si ja existia una altra aresta (v_j, v_k) se suprimeix la de major cost i es manté l'altra.
- Si $c(a_i) > c(C_{ij})$, essent $ai = (v_i, v_j) \Rightarrow$ ai s'elimina.
- Si $v_i \in G$, $v_i \in Z$ i $grau(v_i) = 1 \Rightarrow v_i$ i (v_i, v_j) formaran part de l'arbre de Steiner per força. Aleshores v_i podrà ser eliminat i v_j, considerat $v_j \in Z$ tant si anteriorment $v_j \in Z$ com si no.
- Si $\{v_i, v_j, v_k\} \in G$, $\{v_i, v_j, v_k\} \in Z$, $v_i \in \Gamma(v_j)$ i $c(v_i, v_j) \geq c(C_{ki})$ i $c(v_i, v_j) \geq c(C_{kj}) \Rightarrow (v_i, v_j)$ s'eliminarà de G.

Un cop realitzada la supressió de vèrtexs i arestes prescindibles en G es procedeix a aplicar l'algorisme de Sullivan. El resultat obtingut en l'algorisme és un arbre de cost mínim molt aproximat a l'arbre de Steiner òptim.

2.3 $|Z| = 2$

Quan l'arbre de cost mínim que es vol buscar conté tant sols 2 vèrtexs es pot parlar de camí de cost mínim o camí òptim perquè, en definitiva, no deixa de ser una seqüència ordenada d'arestes o arcs adjacents.

Segons les característiques que presenti el graf s'utilitzarà un o altre algorisme:

- Moore: L'algorisme de Moore s'utilitzarà a l'hora de buscar un camí òptim Cij en un graf simètric G(V, A) els costos de les arestes del qual siguin tots iguals entre ells.
- Dijkstra: L'algorisme de Dijkstra s'utilitzarà per a trobar un camí òptim Cij en un graf dirigit G(V, A) on els costos dels arcs no són negatius. Aquest algorisme també es pot aplicar en grafs simètrics. En aquest cas tan sols caldrà desdoblar cada aresta en dos arcs -un en cada sentit- i, a partir d'aquí, procedir igual que en el cas dels grafs dirigits.
- Ford: L'algorisme de Ford s'aplicarà en grafs dirigits amb costos no restringits en els arcs i sense circuits de cost negatiu. En aquest cas, a diferència de Dijkstra, no es pot aplicar l'algorisme de Ford a grafs simètrics perquè el desdoblament d'arestes de cost negatiu en arcs donaria lloc a circuits de cost negatiu i això bloquejaria l'algorisme.
- Floyd: Aquest algorisme s'aplicarà en grafs dirigits amb costos no restringits en les arestes. En aquest procés és fàcil de detectar la presència de circuits de cost negatiu i, per tant, aquests deixen de ser un problema.

Arborescències

Una arborescència d'un graf dirigit G és un graf parcial de G, $T(V_T, A_T)$ (amb el graf subjacent connex) tal que tot $v_i \in V_T$ té $|\Gamma(v_i)| = 1$ excepte un vèrtex v_r anomenat arrel que té $|\Gamma(v_r)| = 0$.

Si $V_T = V \Leftrightarrow$ l'arborescència és màxima.

2.1.2.4 Camins i circuits

Ens centrarem especialment en dos tipus de camins i circuits: els eulerians i els hamiltonians:

Camins i circuits eulerians

Es designen amb el nom d'eulerians els camins i/o circuits que recorren un graf passant una sola vegada per cadascuna de les seves arestes (arcs).
Reben aquest nom en memòria d'Euler i el problema que plantejà.

1. Existència de camins i/o circuits eulerians.

No tots els grafs presenten un circuit o camí eulerià. Per saber-ho cal recórrer a una sèrie de teoremes que caracteritzen els grafs que en tenen.

- Teorema 1: Un multigraf simètric i connex G(V,A) té camí (circuit) eulerià $\Leftrightarrow$ té com a màxim 2 (0) vèrtexs de grau senar.
- Corol·lari 2: Un graf simètric i connex G(V,A) té circuit eulerià $\Leftrightarrow$ és possible una partició d'A obtenint k circuits disjunts d'arestes.
- Teorema 3: Un graf simètric i connex G(V, A) amb 2k vèrtexs de grau senar $\Rightarrow$ és possible una partició d'A en k camins disjunts d'arestes.
- Teorema 4: Un multigraf dirigit G(V, A) amb el graf subjacent connex té circuit eulerià $\Leftrightarrow$ grau interior(v_i) = grau exterior(v_i) $\forall\ v_i \in V$.
  Un multigraf dirigit G(V, A) amb el graf subjacent connex té camí eulerià de s a t $\Leftrightarrow$ - grau interior(v_i) = grau exterior(v_i) $\forall\ t \neq v_i \neq s$
  - grau interior(t) = grau exterior(t) + 1
  - grau interior(s) = grau exterior(s) – 1

2. Obtenció de circuits eulerians.

Un cop vist si un graf conté un circuit o camí eulerià cal procedir a trobar-lo.

- Obtenció: Es pot aconseguir un circuit eulerià d'un graf mitjançant la regla de Fleury. Aquesta consisteix en anar traçant un camí simple a partir d'un vèrtex qualsevol a la vegada que es van esborrant les arestes (arcs) per les quals es passa. S'ha de tenir en compte únicament de no deixar disconnex el graf resultant cada cop que s'esborri una aresta.

Existeixen també els circuits eulerians mixtos, és a dir aquells que apareixen en grafs mixtos. Per entendre el mecanisme de recerca cal primerament fer un incís en l'obtenció d'un circuit eulerià en grafs mixtos balancejats formats únicament per arcs (és a dir, grafs dirigits els vèrtexs dels quals compleixin que el seu grau interior coincideixi amb l'exterior.)

- Obtenció d'un circuit eulerià en un graf mixt balancejat G on només apareguin arcs: Cal numerar els arcs que surten de cada vèrtex v_i amb l'única condició que l'últim arc numerat de cada vèrtex sigui l'arc que intervingui en l'arborescència màxima de G. Posteriorment cal anar traçant un camí partint de l'arrel i agafant en cada vèrtex l'arc de sortida amb menor numeració. Cal anar seguint aquest procés fins esgotar tots els arcs. El resultat obtingut serà un circuit eulerià.

L'obtenció de circuits eulerians en grafs mixtos balancejats parells és molt semblant a l'anterior.

- Obtenció d'un circuit eulerià mixt: Per poder aconseguir un circuit eulerià en aquest tipus de grafs cal orientar les arestes. Posteriorment se'ls aplica un algorisme basat en l'extensió d'una arborescència T i l'exploració de vèrtexs. L'estenció de l'arborescència es duu a terme a partir d'un vèrtex v_j tal que $v_j \in T$. S'haurà d'utilitzar un arc (v_i, v_j) –on v_i era un vèrtex nou – i v_i passarà a formar part de T, a ser un vèrtex usat i, a la vegada, passarà a ser el vèrtex actual a partir del qual es repetirà el mateix procediment aplicat anteriorment a v_j. L'exploració d'un vèrtex $v_j \in T$ es duu a terme buscant entre $|\Gamma(v_j)|$ vèrtexs nous que intervinguin en arestes. Si v_i és un vèrtex nou adjacent a v_j per l'unió entre els dos d'una aresta, aquesta prendrà el sentit (v_i, v_j) i vi passarà a ser un vèrtex usat, el vèrtex a explorar i $T := T \cup \{v_i\}$.

3. Seqüències de De Bruijn.

Una aplicació o cas particular dels circuits eulerians rau en les seqüències de De Bruijn.

Essent: $\sum$ l'alfabet

σ el nombre de símbols de $\sum$

n la longitud de les paraules formades sobre $\sum$

σ^n el nombre de paraules diferents de n longitud formades en $\sum$.

Una seqüència de De Bruijn és una successió de longitud σ^n formada per σ símbols pertanyents a $\sum$ distribuïts de tal manera que traslladant-los circularment d'un en un s'obtindran les σ^n paraules de longitud n possibles sobre $\sum$.

Qualsevol seqüència de De Bruijn pot estar associada a un graf dirigit $G_{\sigma,n}(V, A)$. Aquest rebrà el nom de diagrama de De Bruijn.

$G_{\sigma,n}(V, A)$ V conjunt de σ^{n-1} paraules de $\sum$ de longitud n – 1

A conjunt de σ^n paraules de $\sum$ de longitud n

Cada arc sortirà d'un vèrtex l'últim símbol de la paraula del qual coincideixi amb el primer de la paraula assignada al vèrtex successor. D'aquesta manera la paraula corresponent a cada arc vindrà donada pel fet de solapar la paraula del vèrtex antecessor amb la del successor.

Per poder obtenir seqüències de De Bruijn només caldrà cercar un circuit eulerià de $G_{\sigma,n}$ i agafar el primer símbol de cada arc del circuit.

Exemple:

$\sum = \{0, 1, 2\}$

$\sigma = 3$

$n = 3$

$\sigma^n = 3^3 = 27$

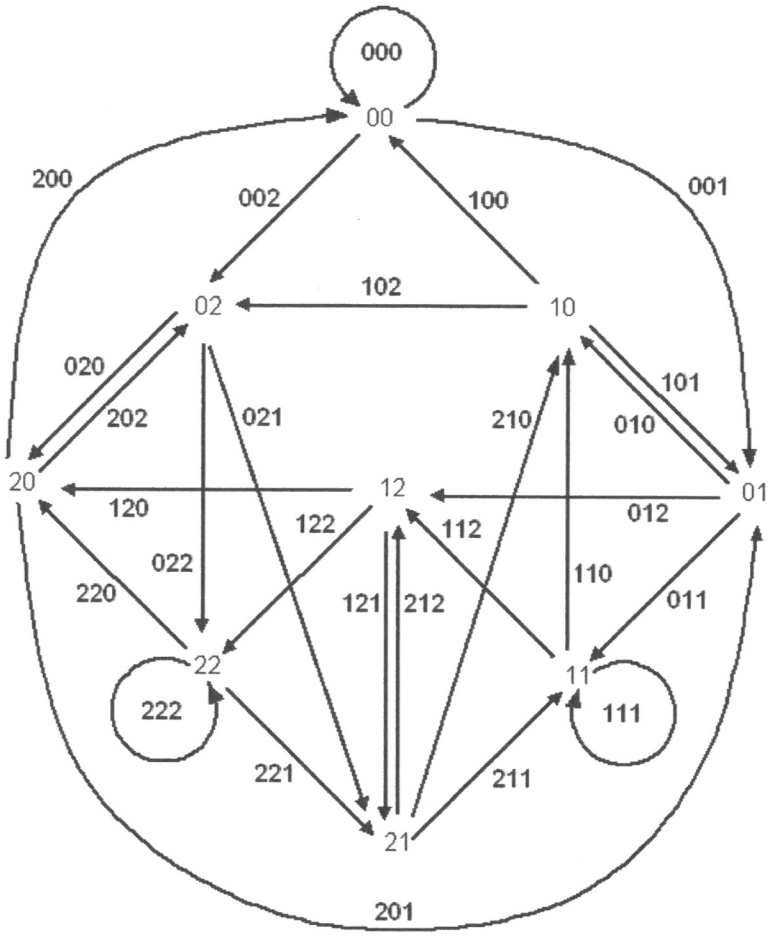
Una possible seqüència de De Bruijn d'acord amb les dades anteriors és:

000100222122021112102011012

S'observa que si es desplaça de manera circular el primer símbol de la seqüència, els tres primers nombres aniran variant donant totes les combinacions possibles (paraules) entre els elements de $\sum$.

0	0	0	1	0	0	2	2	2	1	2	2	0	2	1	1	1	2	1	0	2	0	1	1	0	1	2
0	0	0																								
	0	0	1																							
		0	1	0																						
			1	0	0																					
				0	0	2																				
					0	2	2																			
						2	2	2																		
							2	2	1																	
								2	1	2																
									1	2	2															
										2	2	0														
											2	0	2													
												0	2	1												
													2	1	1											
														1	1	1										
															1	1	2									
																1	2	1								
																	2	1	0							
																		1	0	2						
																			0	2	0					
																				2	0	1				
																					0	1	1			
																						1	1	0		
																							1	0	1	
																								0	1	2
																									1	2	0
																										2	0	0

El diagrama de De Bruijn corresponent a aquesta seqüència és $G_{3,3}$.



4. Problema.

És conegut el problema del carter xinès o CPP (*Chinese Postman Problem*) vinculat als camins i circuits eulerians. Aquest presenta un carter que té uns carrers assignats pels quals ha d'anar repartint cartes. Quin seria el recorregut que hauria de fer per tal de passar per tots els carrers i que aquest fos mínim?

Camins i circuits hamiltonians

S'atribueix l'adjectiu hamiltonià als camins i/o circuits que recorren un graf passant una sola vegada per cadascun dels seus vèrtexs.

El nom prové de Hamilton que els introduí a través del problema del dodecaedre.

Tot circuit hamiltonià és un 2-factor. Un k-factor en un graf G és un subgraf k-regular de G aconseguit a força de suprimir arestes de G.

1. Existència de camins i/o circuits hamiltonians

A l'igual que amb els circuits i camins eulerians, no tots els grafs contindran camins i/o circuits hamiltonians. Per identificar quins d'ells en tinguin caldrà seguir una sèrie de teoremes i corol·laris referents a les característiques que compartiran els grafs hamiltonians.

- Teorema 1: Un graf pla G(V,A) que contingui un circuit hamiltonià H verificarà:

$$\sum_{j>2}(j-2)(R-R')=0$$

  Essent: R el nombre de regions de G interiors a H
  R' el nombre de regions de G exteriors a H
  j el nombre d'arestes que delimiten cada regió.

- Teorema 2: Si un graf simètric G(V, A) és hamiltonià $\Rightarrow C(G-V') \leq |V'|$

  Essent: $V' \in V$ i $V' \neq 0$
  C(G – V') el nombre de components del graf G – V'

- Teorema 3: Tot graf G(V, A) simètric i complet té circuit hamiltonià.
- Teorema 4: Si un graf simètric i connex G(V, A) compleix que $|V| = n > 2$ i grau $(v_i) = n/2$, essent $v_i \in V \Rightarrow$ G és hamiltonià.
- Teorema 5: Si un graf simètric G(V, A) té $|V| \geq 3$ i compleix que per tots els vèrtexs no adjacents v_i i v_j grau(v_i) + grau $(v_j) \geq |V| \Rightarrow$ G és hamiltonià.
- Teorema 6: Si un graf dirigit té el graf subjacent complet $\Rightarrow$ té camí hamiltonià.
- Teorema 7: Si un graf dirigit és fortament connex i amb el graf subjacent complet $\Rightarrow$ té circuit hamiltonià.

2. Obtenció de camins hamiltonians

A continuació es presenten dos mètodes de recerca de camins hamiltonians en grafs dirigits:

- El mètode de les multiplicacions llatines

- El mètode de Roberts i Flores: aquest mètode es basa en el backtracking. Sigui G(V, A) el graf a estudiar cal elaborar una taula P de k files i n columnes. A la primera fila s'hi col·locarà cadascun dels vèrtexs $v \in V$. A continuació sota v_i s'hi col·locarà el $\Gamma(v_i)$ omplint la columna v_i; sota v_j caldrà posar-hi $\Gamma(v_j)$... i així fins completar totes les columnes corresponents a cadascun dels vèrtexs de G. D'aquesta manera s'haurà obtingut una taula on $n = |V|$ i $k = \max_{v_i \in V} |\Gamma(v_i)|$. Un cop feta la taula s'iniciarà l'elaboració d'una llista S que contindrà tots els vèrtexs pels que s'anirà passant i, si G és hamiltonià donarà el camí o circuit que es busca. Per obtenir S cal fer el següent:

  S'agafa el vèrtex v_i corresponent a la primera columna i s'afegeix a S. A continuació es busca el primer vèrtex $v_j \in \Gamma(v_i)$ no usat i s'incorpora a S. Després cal buscar el primer vèrtex no usat $v_k \in \Gamma(v_j)$ i així successivament.

  Suposem que $S = \{v_i, v_j, v_k, \ldots, v_{r-1}, v_r\}$

  Al final pot passar:

  a. Que no hi hagi cap vèrtex no usat en la columna on s'estigui actualment buscant i no apareguin tots els vèrtexs $v \in V$ en S.

  b. Que no existeixi cap vèrtex no usat en la columna on s'està buscant actualment i que apareguin en S tots els vèrtexs $v \in V$.

     b.1 Que $v_i \notin \Gamma(v_r)$

     b.2 Que $v_i \in \Gamma(v_r)$

Segons quin sigui el cas de G ens trobem amb:

Cas		Resposta
a		Caldrà fer backtracking
b	1	S → camí hamiltonià
	2	S → circuit hamiltonià

En el cas b, S contindrà un camí hamiltonià. Només caldrà buscar si la columna en qüestió conté el primer vèrtex que aparegui en S de manera que, en cas afirmatiu, en comptes de camí, G tindrà un circuit hamiltonià. En el cas a, en canvi, caldrà fer backtracking.

El backtracking és un procediment que consisteix en, un cop tallat el camí per l'absència de vèrtexs no usats en la columna en la qual s'està buscant, retrocedir a l'anterior columna i cercar-hi un altre vèrtex no usat de manera que permeti continuar el camí per una altra via. És a dir, si tenim $S=\{v_1, v_i, v_j, v_k\}$ i $\{v_1, v_i, v_j, v_k\}\in V$ essent V G(V, A) i G consta de més de 4 vèrtexs s'haurà de retrocedir de la columna de v_k a la de v_j i buscar-hi un vèrtex $v \neq v_k$ i continuar a partir de v. Si $v\in\Gamma(v_j) \Rightarrow$ caldrà fer backtracking a v_i i així successivament fins arribar a v_1. El procés acabarà quan no hi hagi $v\in\Gamma(v_1)$ possible per a continuar.

3. Problema

Dins de l'apartat de camins i circuits hamiltonians s'estudia el problema del viatjant o TSP (traveling salesman problem). Aquest presenta un viatjant que ha de passar per uns llocs determinats. Quin serà el recorregut mínim que haurà de fer per passar per cada indret el menor nombre de vegades possible sense descuidar-se'n cap?

2.1.2.5 Planaritat

Un graf és pla quan és possible la seva representació gràfica de manera que no es tallin les arestes en el pla.

Les diferents zones en que ha quedat dividit el pla s'anomenen regions i venen delimitades per les arestes del graf. La regió exterior és aquella que no queda limitada per les arestes de manera que resta envoltant el graf.

Un graf pla extern és aquell tots els vèrtex del qual estan en contacte amb la regió exterior.

Caracterització:

Els grafs plans comparteixen unes mateixes característiques. Els següents teoremes i corol·laris permetran identificar o distingir d'altres que no ho siguin els grafs plans.

- Teorema: Tot graf pla i connex verifica la fórmula d'Euler.

La fórmula d'Euler diu que, essent G(V, A) un graf connex amb R regions es complirà:

$$|R| + |V| = |A| + 2$$

- Corol·lari: Tot graf pla, connex i sense llaços G(V, A) on $|V| \geq 3$ compleix:

$$3|V| - 6 \geq |A|$$

- Corol·lari: Tot graf pla, connex i bipartit G(V, A) on $|A| \geq 3$ compleix:

$$|A| \leq 2\,|V| - 4$$

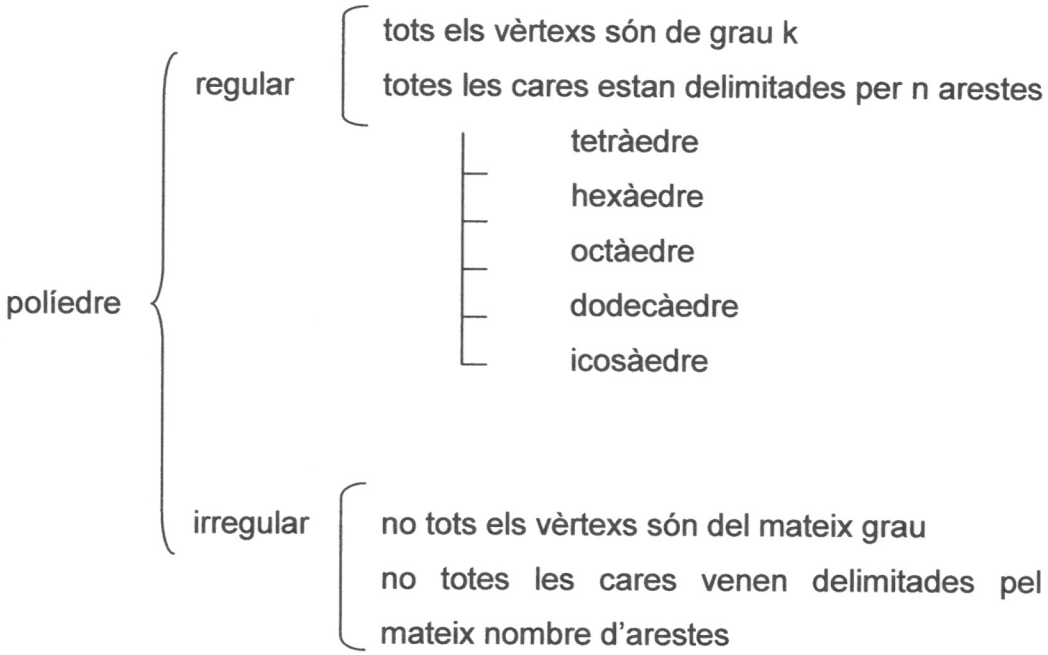
- Corol·lari: Tot graf pla G(V, A) amb $n \geq 3$ vèrtexs i essent n_i el nombre de vèrtexs de grau i, per i = 1, 2, ..., m, compleix:

$$5n_1 + 4n_2 + 3n_3 + 2n_4 + n_5 \geq n_7 + 2n_8 + \ldots + (m-6)n_m + 12$$

- Corol·lari: Tot graf pla conté almenys un vèrtex v_i grau(v_i)< 6

Els políedres

Un políedre és un cos geomètric les cares del qual són polígons. Un políedre pot ser:



Els políedres es poden representar mitjançant un graf planar. Les arestes i vèrtexs del políedre coincidiran amb les arestes i vèrtexs del graf mentre que les cares passaran a ser les regions del graf (incloent-hi la regió externa). Tant els políedres com els grafs planars que els corresponen verifiquen la fórmula d'Euler.

2.1.2.6 Coloració

El tema de la coloració intervé en els grafs a dos nivells:

- **Coloració de vèrtexs**

Aquest tipus de coloració consisteix en atribuir a cada vèrtex v_i un color x de manera que cap vèrtex $v_j \in \Gamma(v_i)$ tingui x assignat. El vèrtex $v_i \in V$, essent G(V, A) un graf pla, simètric i sense llaços. El nombre de colors a emprar r ha de ser mínim. En aquest cas r s'anomenarà nombre cromàtic de G i es representarà $\gamma(G)$ i G serà r-colorable.

Els valors de $\gamma(G)$ màxim i mínim –exceptuant en els casos on G és trivial– es produiran en els següents casos:

Valor màxim ⟶ graf complet ⟶ $\gamma(G) = n$

Valor mínim ⟶ graf bipartit ⟶ $\gamma(G) = 2$

No es coneix cap algorisme per determinar el $\gamma(G)$ òptim. No obstant, existeixen una sèrie de teoremes i regles que permeten conèixer $\gamma(G)$ per a segons quins tipus de grafs:

- G(V, A) és trivial $\Rightarrow \gamma(G) = 1$.
- Teorema: G és 2-colorable $\Leftrightarrow$ no conté cap circuit amb nombre senar d'arestes.
- Teorema: Tot graf pla G(V, A) té $\gamma(G) \leq 5$.

De fet, si que existeix un algorisme que permet colorar un graf pla utilitzant 5 colors però, així com van demostrar Appel i Haken que el $\gamma(G)$ òptim per a qualsevol graf pla és 4, no es coneix un algorisme eficient per a grafs 4-colorables.

- Si G(V, A) és pla i té un nombre de circuits de 3 arestes de longitud < 4 $\Rightarrow$ G és 3-colorable.

- **Coloració d'arestes**

Aquest tipus de coloració consisteix en assignar a cada aresta (v_i, v_j) un color x de manera que cap aresta (v_i, v_k) o (v_k, v_j) tingui x associat. Això s'ha de fer utilitzant el menor nombre de colors possible.

$\gamma_a(G)$ és el nombre cromàtic per a les arestes de G i representa el menor nombre de colors a utilitzar per colorar les arestes de G possible.

$$\gamma_a(G) \geq \Delta(G)$$

Els següents teoremes permeten trobar $\gamma_a(G)$ per segons quins tipus de graf o, si més no, ajuden a precisar-ne el valor:

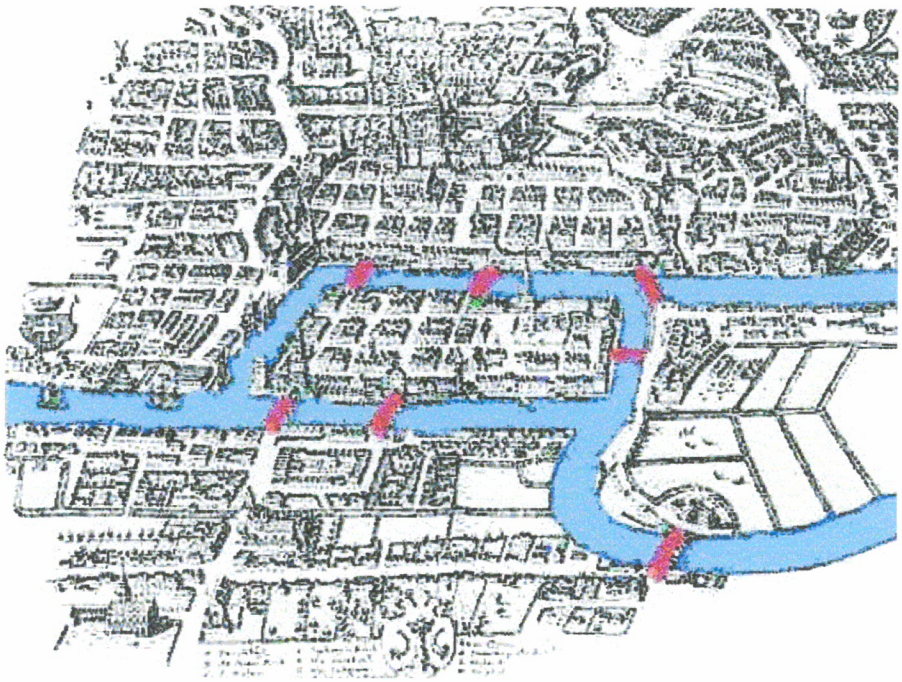
- Teorema: Si G(V, A) és bipartit $\Rightarrow \gamma_a(G) = \Delta(G)$
- Teorema: Si G(V, A) és K_n tal que n és senar $\Rightarrow \gamma_a(G) = \Delta(G) + 1$

  Si G(V, A) és K_n tal que n és parell $\Rightarrow \gamma_a(G) = \Delta(G)$
- Teorema: Sigui G(V, A) un graf sense llaços $\Rightarrow \Delta(G) \leq \gamma_a(G) \leq \Delta(G) + 1$

2.1.3 Problemes

- Els ponts de Königsberg

Aquest problema sorgí arran de les passejades que feien els habitants de Königsberg (avui, Kaliningrad). El riu Pregel travessava la ciutat deixant al mig una illa anomenada Kneiphof (dib. 1).

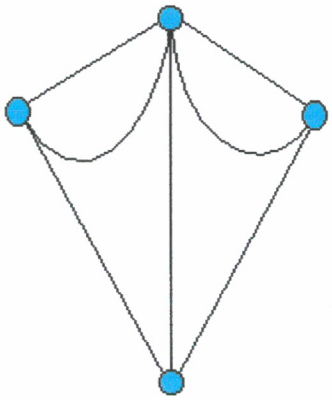


Dibuix 1

Set ponts permetien creuar el riu per passar d'unes ribes a les altres. Els ciutadans de Königsberg es qüestionaven si seria possible fer una passejada passant un únic cop per cada pont i acabant el recorregut en el mateix punt de partida. Mentre hi havia qui dubtava hi havia d'altres que ho negaven malgrat tot, no fou fins que Euler abordà el problema que es trobà una explicació convincent que demostrava la impossibilitat de realitzar un recorregut amb aquestes característiques. Euler, a més a més, generalitzà la demostració (publicada al 1736 en un treball sobre el problema). Aquesta és la que avui dia es coneix amb el cèlebre nom de: teorema d'Euler.

És a partir d'aquest problema que nasqué la teoria de grafs ja que, tot i que sense saber-ho encara, Euler transformà el conjunt de ribes i ponts en graf

passant aquests a ser vèrtexs i arestes respectivament. Com que cap vèrtex resultava ser parell esdevenia impossible la determinació d'un circuit eulerià per al graf en qüestió.



Dibuix 2

Resolució:

Al resultar un graf no eulerià és impossible recórrer un sol cop tots els ponts acabant en el mateix punt de partida. Es pot buscar però un recorregut que permeti acabar al mateix punt d'inici recorrent el mínim de cops possible cada pont i passant almenys una vegada per cadascun d'ells. Aquesta solució es pot trobar mitjançant Edmonds.

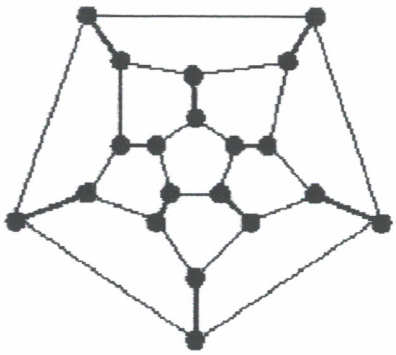
- El dodecàedre de Hamilton

Aquest problema fou plantejat per Hamilton al 1859. El matemàtic considerava cada vèrtex d'un dodecàedre una ciutat important i proposava de realitzar un viatge per les arestes per visitar cada ciutat un sol cop.

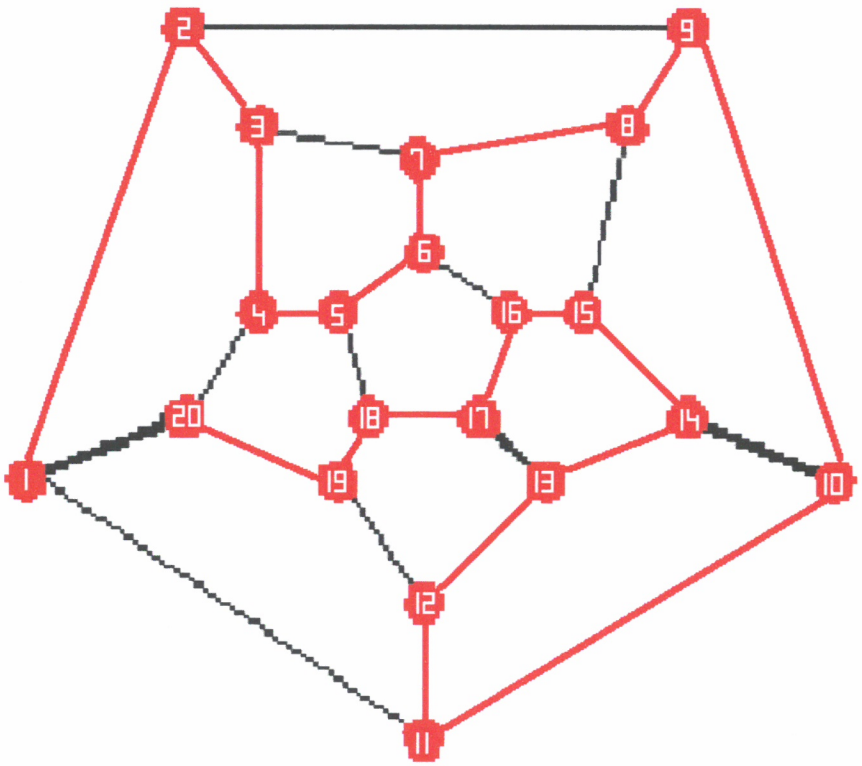
Resolució:

La resposta a aquest problema es troba a partir de la transformació de la figura en graf. D'aquesta manera s'obté un graf planar (dib. 4) sobre el qual es pot aplicar el mètode de Roberts i Flores per a trobar el circuit hamiltonià solució (dib. 5).

Dibuix 3



Dibuix 4



Dibuix 5

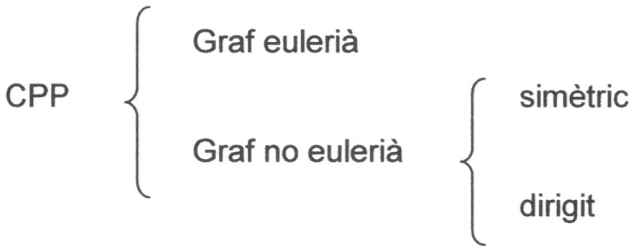
- El problema del carter xinès

Aquest problema va ser plantejat pel xinès Kwan Mei-ko. Diu:

> "Un carter s'encarrega de repartir la correspondència d'una zona determinada. Quin recorregut haurà de fer per tal de passar per tots els carrers –a ser possible un cop com a mínim per cadascun– de manera que aquest sigui mínim?"

Resolució:

La resolució del problema es basa en l'existència d'un camí o circuit eulerià present en el graf format a partir dels carrers considerats com a arestes. Ara bé pot passar que el graf obtingut no sigui eulerià. Per tant podem classificar el problema de la manera següent en funció de com sigui el graf:



Caldrà, a més a més associar un cost c_{ij} a cada aresta en funció, per exemple, de la longitud del carrer.

1. CPP en un graf eulerià

La resposta al problema coincidirà amb el camí o circuit present en G.

2. CPP en un graf no eulerià simètric

La resposta al problema es pot trobar mitjançant el mètode Edmonds

3. CPP en un graf no eulerià dirigit

La resposta en aquest cas ja és més complexa, si més no, la manera de trobar-la ja que hi intervenen nocions de fluxos i de xarxes de transport.

- El problema del viatjant

Aquest problema diu:

> " Un viatjant ha d'anar a visitar una sèrie d'indrets. Quin seria el camí òptim per dur a terme el seu recorregut per tal que hagués de passar el mínim de vegades possible (o un sol cop) per cadascun dels llocs?"

Això traslladat al terreny dels grafs es tradueix convertint els indrets a visitar en els vèrtexs del graf i les carreteres, camins o vies de comunicació entre els llocs en les arestes o arcs del graf. Cada aresta o arc té un cost assignat en funció de la longitud de la via etc... . Dins del TSP hi ha:

TSP { TSP generalitzat
TSP { TSP no generalitzat

El TSP generalitzat demana que es passi el mínim de vegades possible per cada vèrtex i el TSP no generalitzat, que s'hi passi un sol cop per cadascun.

Sigui G = K_n G contindrà: $\frac{n!}{2}$ camins hamiltonians

$\frac{(n-1)!}{2}$ circuits hamiltonians

Resolució:

El mètode a utilitzar variarà segons la recerca sigui d'un camí o circuit hamiltonià de cost mínim (en grafs G = K_n):

recerca	mètode a utilitzar
1. camí hamiltonià	1.1 Kruskal i eliminació d'arestes
2. circuit hamiltonià	2.1 mètode basat en l'arbre de cost mínim
	2.2 mètode basat en l'aparellament perfecte òptim

1. Camí hamiltonià òptim

La recerca d'un camí hamiltonià òptim és fa entre un arbre de possibilitats i en base al:

Teorema: Sigui G(V, A) un graf simètric i T(G) un arbre generador per a G tal que tot $v_i \in V$ grau $(v_i) \leq 2 \Rightarrow$ T(G) és un camí hamiltonià.

Per obtenir T(G) cal aplicar a G l'algorisme de Kruskal. Ara bé, pot passar que l'arbre obtingut T(G) tingui un vèrtex v_i el grau del qual sigui $n > 2$ i, per tant T(G) no serà hamiltonià. Aleshores s'haurà de buscar n arbres generadors per a G: $T_1(G)$, $T_2(G)$, ..., $T_n(G)$ on en cadascun d'ells s'haurà eliminat prèviament una de les arestes (v_i, v_j). Anomenarem $T_{1,n}(G) = \{T_1(G), T_2(G), ..., T_n(G)\}$. Aquesta eliminació es fa considerant que $c_{vi, vj} = \infty$ de manera que, en cada cas $(T_2(G), T_3(G), ..., T_n(G))$ caldrà buscar la següent aresta de cost mínim per obtenir el nou arbre generador de G. És a dir, s'haurà de tornar a aplicar Kruskal considerant en cada cas $c_{vi, vj} = \infty$. Un cop realitzat aquest pas caldrà calcular els costos dels arbres obtinguts i mirar si tots ells són camins hamiltonians. Si resulta haver-hi un arbre $T_m(G) \in T_{1,n}(G)$ no hamiltonià tal que $c(T_m) < c(T_k)$ essent $T_k(G) \in T_{1,n}(G) - T_m(G)$ caldrà aplicar a $T_m(G)$ el procés seguit anteriorment en el vèrtex $v_k \in T_m(G)$ tal que grau $(v_k) > 2$ fins arribar a una solució òptima. Si $T_{1,n}(G)$ són hamiltonians la solució serà l'arbre generador amb un menor cost associat.

2. Circuit hamiltonià òptim

Per a trobar el circuit hamiltonià d'un graf $G = K_n$ es podrà recórrer a dos algorismes que permetran trobar un circuit hamiltonià H molt aproximat al circuit hamiltonià òptim de G, H_0.

2.1 El mètode basat en l'arbre de cost mínim

Cal buscar l'arbre generador de cost mínim per a G mitjançant Kruskal. Un cop trobat, s'haurà de determinar la seqüència de vèrtexs a seguir com a resultat d'aplicar el DFS a l'arbre generador T(G). Finalment es numerarà cada k-èsim vèrtex nou v_i de la llista obtinguda amb una k. El circuit hamiltonià aproximat a

H_0 estarà definit per la seqüència de vèrtexs ordenats segons la numeració feta prèviament.

2.2 El mètode basat en l'aparellament perfecte òptim

Cal buscar també per Kruskal l'arbre generador de cost mínim T(G) per a G. En segon lloc s'haurà de fer una llista dels vèrtexs v_i de grau senar. A partir d'aquests vèrtexs es buscarà l'aparellament perfecte òptim M* dins el graf K_m format a partir dels m vi trobats en T(G). Per acabar caldrà duplicar les arestes de M* en T(G) amb la qual cosa s'haurà obtingut un circuit eulerià. Finalment s'etiquetarà cada k-èssim vèrtex v_i de la seqüència que definirà el circuit eulerià amb una k. El resultat serà una llista de tots els vèrtexs de T(G) ordenats segons el valor de k assignat a cadascun que constituirà un circuit hamiltonià H.

- Recorregut d'un laberint — Visita al laberint d'Horta.

El laberint d'Horta forma part dels jardins històrics que porten el mateix nom i que es troben al districte d'Horta-Guinardó de Barcelona. Aquests van ser construïts a finals de s. XVIII per Doménico Bagutti entre d'altres arquitectes i fins a finals de s. XX van pertànyer al Marquès de Llupià i d'Alfarràs. Els jardins estan distribuïts en tres nivells: el més elevat conté un safareig des d'on s'administra l'aigua destinada al rec, el segon consta d'unes torres-mirador i sota es troba el laberint. El parc està envoltat per una pineda i conté nombroses escultures, pèrgoles...

El laberint està construït a base de xiprers tallats que constitueixen les parets. L'entrada es troba a un extrem. Presenta una inscripció en una taula de pedra i ofereix dos camins a triar. Al centre del laberint hi ha una plaça circular amb bancs de pedra i vuit arcades de xiprer que, formant una circumferència, cobreixen l'inici de cadascun dels camins que en surten. Al bell mig d'aquesta hi ha una estàtua d'Eros. El camí que condueix a la sortida desemboca a una altra plaça amb un estany amb un sortidor.

Un laberint és una estructura no lineal, amb bifurcacions, camins i vies mortes. Això permet traslladar-lo a terreny de grafs. Els trams de camí passaran a ser les arestes i les bifurcacions entre ells o els atzucacs, els vèrtexs. D'aquesta manera es podrà aplicar al laberint un tractament propi dels grafs que permeti recórrer-lo i trobar la sortida sense estancar-se. Així doncs, el problema es presenta de la següent manera:

- Objectiu: arribar a la meta (sortida, plaça central...)
- Procediment: recórrer el laberint amb un cert control dels camins que s'escullen (per evitar possibles pèrdues)

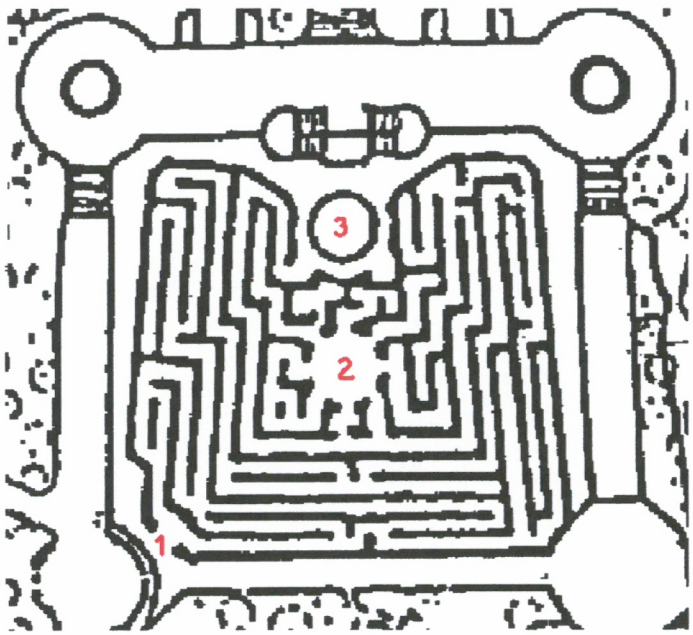
Aquest en termes matemàtics es planteja:

- Objectiu: anar del vèrtex inicial w al vèrtex final f.
- Procediment: recórrer el graf fins arribar a f.

Per recórrer un graf s'ofereixen dos procediments: el BFS i el DFS. Ambdós són arbres amb Z = V i això garanteix l'existència de f dins l'arbre. La naturalesa del BFS però, comporta certes incomoditats a l'hora de resoldre el problema . L'única opció favorable doncs, és l'aplicació del DFS. Aquesta solució obligarà a recórrer una rere altra cada aresta del graf. Al arribar a un vèrtex v_i tal que grau(v_i) = 1 caldrà tornar a v_{i-1} i des d'allí agafar una altra aresta no usada encara per tal de tornar a emprendre el camí. Aquest procediment es coneix a nivell col·loquial com a mètode de la mà dreta (esquerra). Consisteix en escollir en cada bifurcació el camí de la dreta (o esquerra segons s'hagi decidit). En cas d'arribar a un atzucac caldrà tornar a la bifurcació anterior i agafar el següent camí possible no recorregut encara de la dreta. L'estratègia de la mà és sempre vàlida excepte en laberints que presentin un anell de murs al voltant de la meta. En aquests casos, al aplicar el procediment descrit es tornarà sense haver arribat a la meta, al punt de sortida. Això succeeix perquè es crea un circuit que envolta la meta: al seguir l'estratègia de la mà dreta (esquerra) la meta queda a l'esquerra (dreta) quedant així inaccessible. La solució en aquests casos és tornar a l'última bifurcació visitada i prendre des d'allí el següent camí possible. Un cop fet això se seguirà utilitzant l'estratègia de la mà. També hi pot haver circuits que no envoltin la meta. En aquest cas no resultaran un problema a l'hora d'aplicar el DFS.

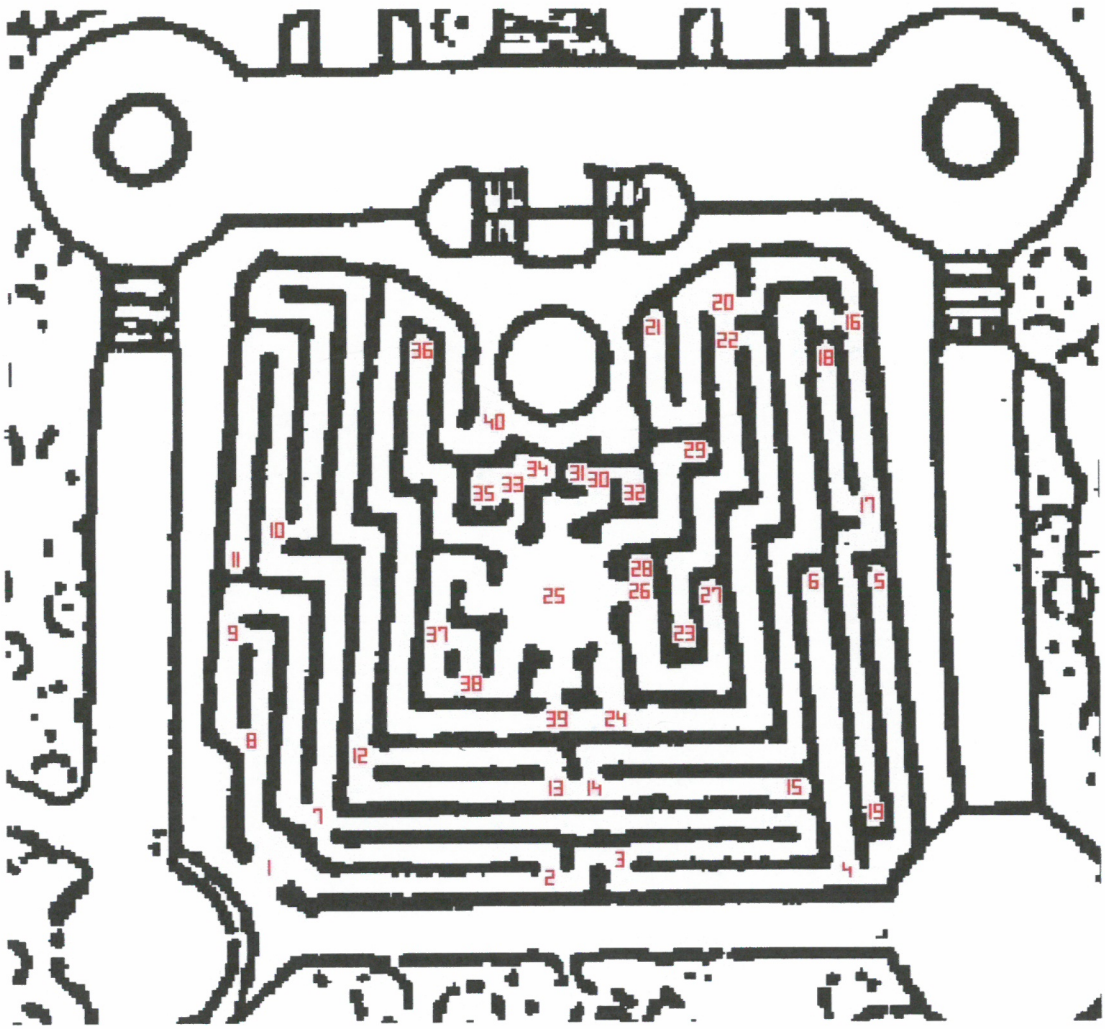
Aquest és el plànol del Laberint d'Horta. (dib. 6)

1 – entrada
2 – plaça central
3 – sortida



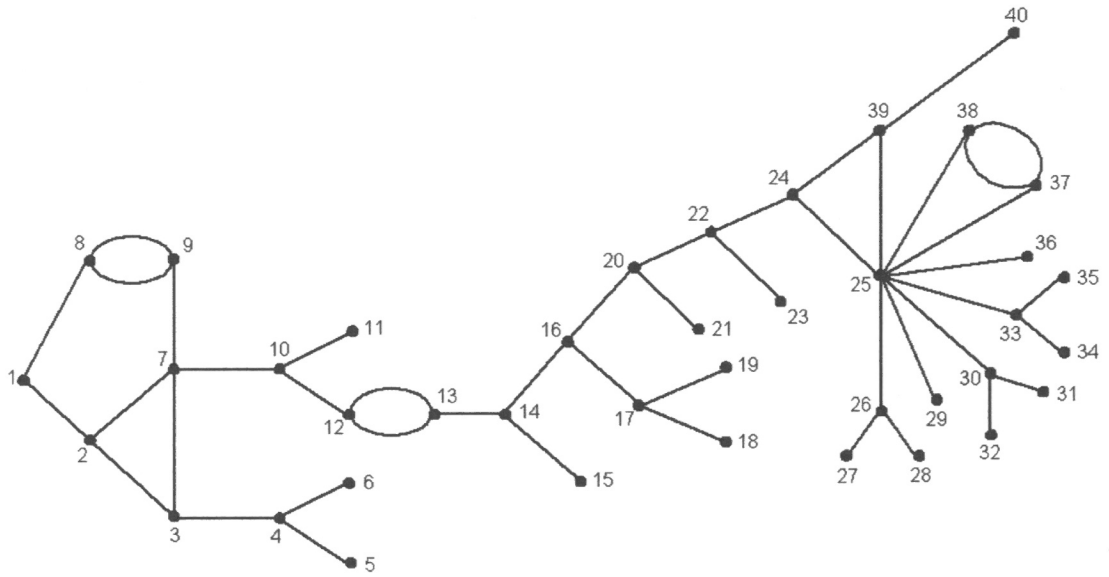
Dibuix 6

Per aplicar el DFS al laberint caldrà numerar les bifurcacions i els atzucacs. (dib. 7)



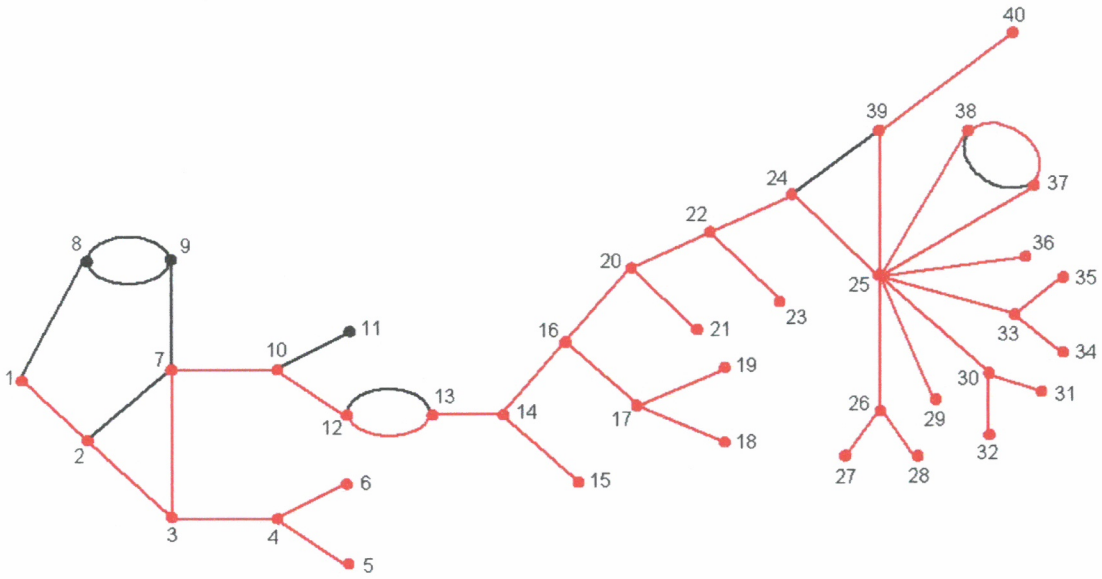
Dibuix 7

Un cop fet això es procedirà a la transformació del laberint en graf prenent les bifurcacions i atzucacs com a vèrtexs i els camins com a arestes. (Graf G_{23})

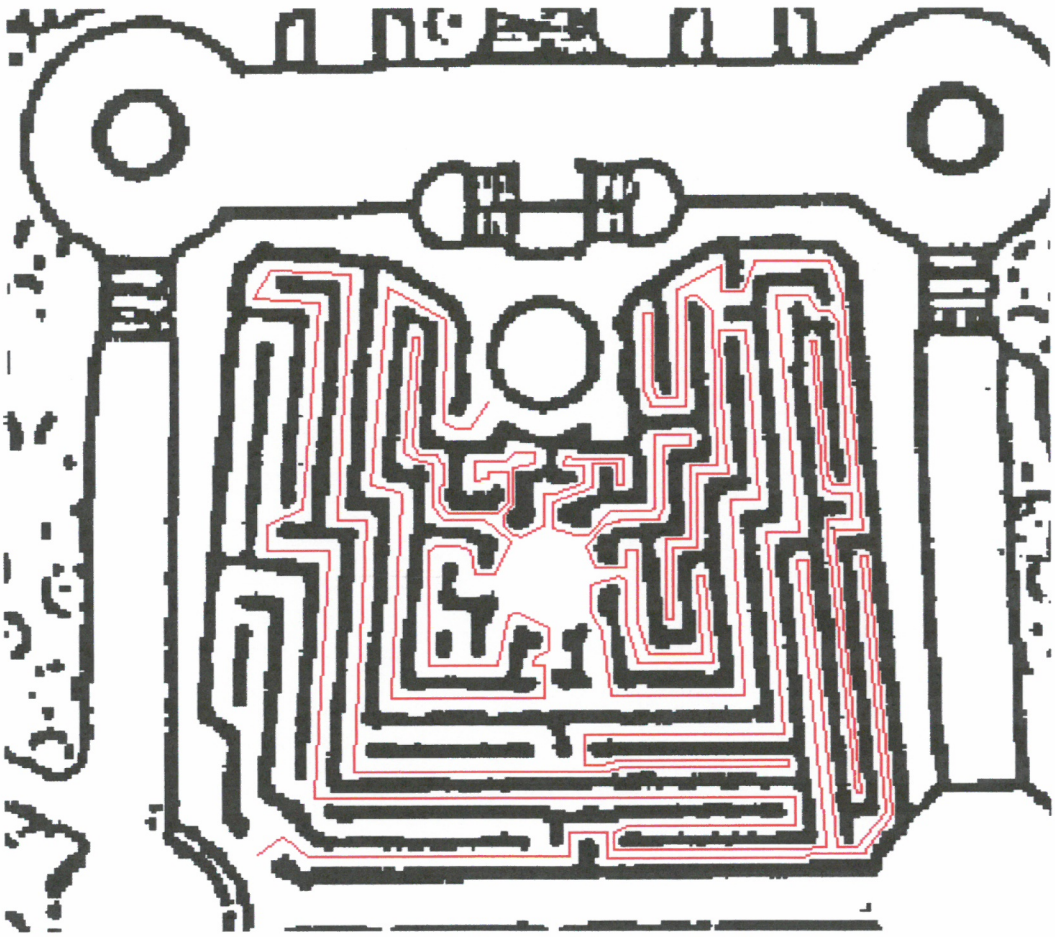


Graf G_{23}

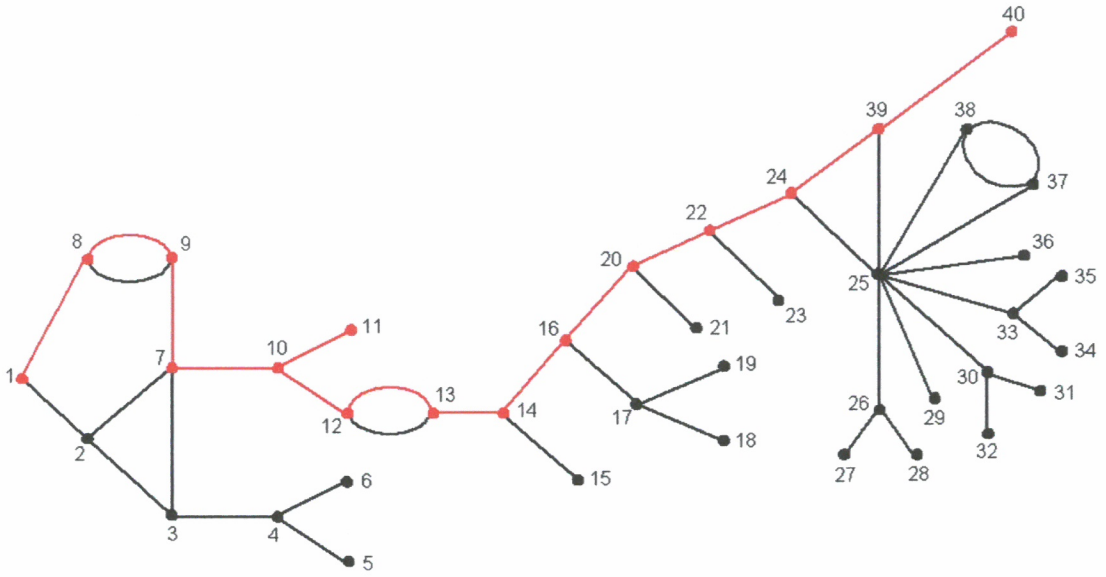
Aplicant al graf l'algorisme DFS s'obtindrà la solució al problema del laberint. Així doncs, la resposta al laberint aplicant "l'estratègia de la mà dreta" serà la que s'indica en el graf G_{23}' i en el (dib. 8) i la resposta obtinguda a partir de "l'estratègia de la mà esquerra" la mostrada en el graf G23'' i en el (dib. 9).



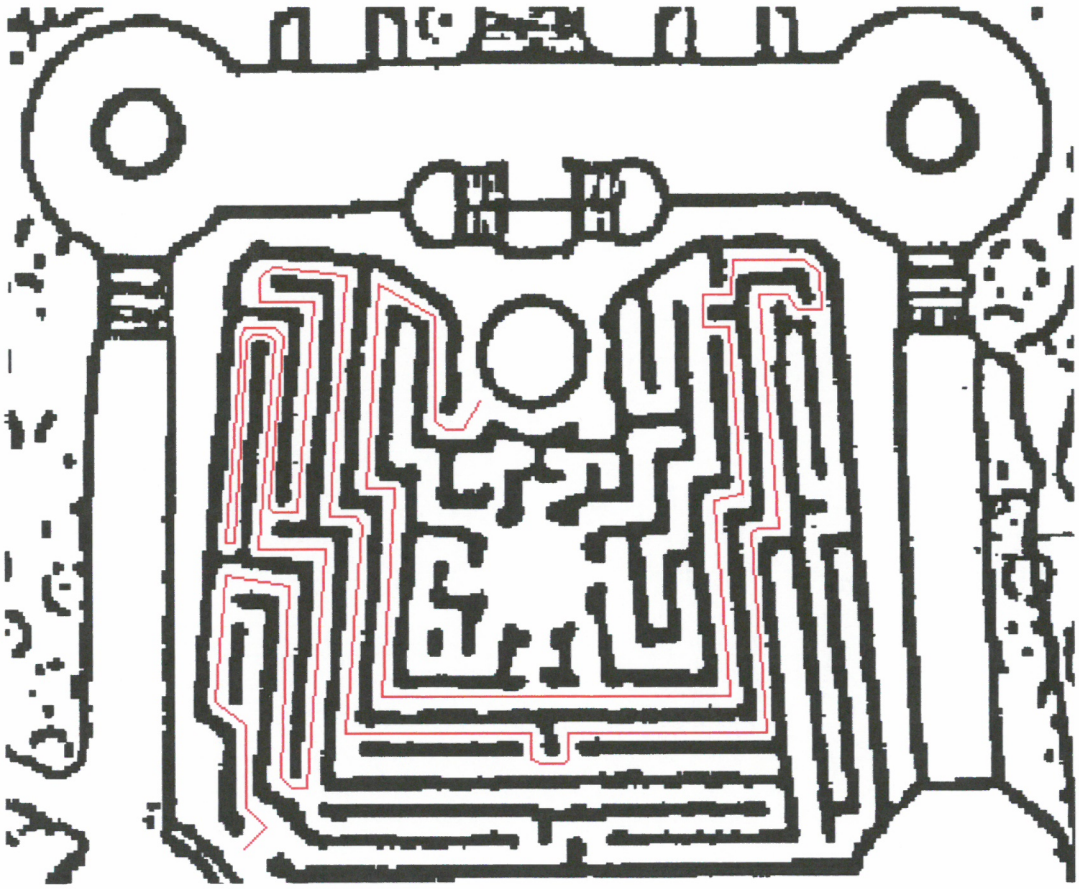
Graf G_{23}'



Dibuix 8



Graf G_{23}"



Dibuix 9

2.2 Pràctica

A continuació es presenten tres experiències d'aplicabilitat d'algun punt de la teoria de grafs a situacions que, en major o menor mesura, poden ben donar-se en el dia a dia. Totes tres estan elaborades sobre el mapa de Catalunya.

(Veure l'execució de "pràctica")

Vols estalviar benzina?

1. **Plantejament del problema**

   Et trobes a Amposta i vols traslladar-te a una capital de comarca de Catalunya concreta. Ara bé, vols que el camí que escullis d'entre les opcions que se't presenten sigui el més curt possible.

2. **Preparació de la pràctica**

   - Cercar un mapa de Catalunya a www.xtec.es
   - Col·locar la imatge en Microsoft Photo Editor, augmentar-la i, amb l'ajuda del cursor, anar trobant les coordenades (en píxels) dels punts que en formen el contorn; és a dir, les coordenades dels punts que constitueixen la frontera de Catalunya amb França, Andorra, Aragó, València i el mar.
   - Trobar les coordenades de les capitals de comarca seguint el mateix procediment.
   - Elaborar una llista ordenada alfabèticament de les 42 capitals de comarca. Cada comarca estarà associada al nombre de posició que ocupi en la llista.
   - Comprovar l'existència d'un camí directe (sense passar per cap altra capital de comarca) entre dues capitals diferents. Per fer-ho, cal anar a www.michelin.es i provar cadascuna de les 861 possibles combinacions entre capitals.
   - Trobar la longitud (en quilòmetres) dels camins obtinguts en el pas anterior.
   - Elaboració d'un graf de 42 vèrtexs (corresponents a les capitals de comarca), les arestes del qual seran els camins trobats

d'antuvi. El cost de cada aresta vindrà donat pel nombre de quilòmetres del camí representat per aquesta.

- Tractament de dades per tal que els vèrtexs del graf estiguin col·locats d'acord amb les coordenades que els corresponen sobre el mapa de Catalunya.

3. **Aplicació**

   Cal aplicar Dijkstra al graf creat.

   (Veure practica/Dijkstra/Dijkstra.exe del CD)

4. **Funcionament**

   La llista de l'esquerra indica la correspondència entre el nombre de vèrtex i la capital. Quan es trobi el camí òptim per arribar a un lloc determinat les arestes que constitueixen aquest camí quedaran marcades en blau i, en la llista, al costat de cada vèrtex, apareixerà el quilometratge. Aquest vindrà donat per la suma de costos associats a les arestes que intervinguin en el camí per arribar des del punt de partida al vèrtex (capital) indicat.

5. **Solució**

   S'obtindrà un arbre l'arrel del qual serà el vèrtex corresponent a Amposta des d'on s'indicarà el camí òptim per arribar a qualsevol altra capital. La llista indicarà al costat de cada capital els quilòmetres mínims per arribar-hi des d'Amposta.

6. **Observacions**

   - Un cop trobat l'arbre solució els vèrtexs quedaran pintats en ordre cromàtic en funció de la proximitat al vèrtex arrel; és a dir, com més curt sigui un camí per anar del punt de partida a un vèrtex, més fosc serà el color d'aquest vèrtex d'arribada.
   - Si en comptes de partir d'Amposta es vol sortir d'una altra capital de comarca només caldrà clicar amb el ratolí el vèrtex del graf des del qual es vol trobar l'arbre.
   - Topològicament una aresta és el mateix que el camí que representa.
   - Alguna aresta surt dels límits del mapa. Això és causat per la unió que s'ha fet entre els vèrtexs utilitzant arestes rectes. En cap cas

simbolitza que el trasllat per anar d'una capital a una altra s'hagi de fer per mar!

Fem turisme!

1. **Plantejament del problema**

   Un estranger arriba a Catalunya. Decideix fer una ruta turística per visitar totes les capitals de comarca i acabar el recorregut en el punt de partida. Per fer-ho lloga un cotxe per poder-se desplaçar. Ara bé, vol saber quin és el trajecte més curt que passi per totes aquestes ciutats.

2. **Preparació de la pràctica**

   La preparació d'aquesta pràctica coincideix totalment amb l'anterior. Per tant, no caldrà buscar ni fer res si abans ja s'ha trobat.

3. **Aplicació**

   Aquest problema és, de fet, el problema del viatjant. Així doncs el tractament que rebrà serà el mateix que el que s'aplicaria a aquest últim.
   (Veure practica/Viatjant/Viatjant.exe del CD)

4. **Funcionament**

   De la mateixa manera que en la pràctica anterior la llista de l'esquerra indica el nombre de vèrtex corresponent a cada capital de comarca. A la dreta, sobre el graf, apareix un circuit hamiltonià inicial de color blau. A partir d'aquí, s'intenta trobar un circuit hamiltonià alternatiu més curt que el que està en pantalla (o bé un circuit hamiltonià alternatiu a partir del qual se'n pugui trobar un de menor quliometratge*). Quan es troba, aquest apareix pintat en el graf. Sobre la llista de la dreta figura l'últim circuit trobat fins al moment amb el quilometratge total. Aquesta informació va variant en funció dels nous circuits que es van trobant.

5. **Solució**

   La solució teòrica d'aquesta experiència és un circuit hamiltonià; és a dir, un recorregut per unes determinades arestes del graf que uneixin tots els vèrtexs (capitals) i que la suma total dels costos de les quals sigui mínima.

6. Observacions

- El problema del viatjant és un problema NP-hard (nondeterministic polynomial). Això significa que no hi ha un algorisme de temps polinomial que el resolgui.
- Com a conseqüència del punt anterior no he pogut arribar a obtenir el circuit hamiltonià òptim.

  Els primers resultats obtinguts han estat:

Circuit	Quilòmetres	Circuit	Quilòmetres
1	2158	17	2032
2	2145	18	2024
3	2138	19	2021
4	2124	20	2013
5	2116	21	2012
6	2136*	22	2011
7	2129	23	2003
8	2115	24	1996
9	2130*	25	1988
10	2110	26	1985
11	2102	27	1977
12	2076	28	1974
13	2056	29	1967
14	2048	30	1959
15	2047	31	1956
16	2039	...	

El circuit que he obtingut més aproximat a l'òptim ha estat el següent amb un total de 1956 km format per la seqüència:

1-9-7-11-15-2-16-5-8-14-13-26-34-28-4-17-27-12-6-10-3-21-39-25-23-29-31-40-22-36-30-33-18-38-42-41-19-37-32-24-35-20-1

Quants en necessitaré?

1. **Plantejament del problema**

   En Francesc arriba a casa.

   – Mare, avui hem estudiat les comarques de Catalunya i les seves capitals a l'escola. La professora ens ha dit que hem de pintar un mapa de manera que les comarques adjacents no rebin un mateix color. Quants colors creus que necessitaré per fer-ho?

2. **Preparació de la pràctica**

   - Cercar a www.xtec.es un mapa polític de Catalunya.
   - Col·locar la imatge en Microsoft Photo Editor, augmentar-la i, amb l'ajuda del cursor, anar trobant les coordenades (en píxels) dels punts que en formen el contorn; és a dir, les coordenades dels punts que constitueixen la frontera de Catalunya amb França, Andorra, Aragó, València i el mar. Trobar també de la mateixa manera les coordenades dels punts que constitueixen les fronteres entre comarques.
   - Trobar a partir del mapa anterior les comarques que delimiten amb cada comarca i elaborar una llista amb la informació obtinguda.
   - Eliminar de la llista una de cada parella de comarques repetida. (Per exemple, si en la llista hi ha apuntades les parelles de comarques veïnes "L'Anoia – La Segarra" i "La Segarra – L'Anoia" se n'haurà d'eliminar una de les dues i mantenir l'altra.)
   - Elaborar un graf de 41 vèrtexs (corresponents a les 41 comarques) les arestes del qual representaran l'existència d'una frontera entre les dues comarques representades pels vèrtexs dels extrems de cadascuna d'elles. Aquestes arestes no tindran cost associat.
   - Tractament de dades per tal que els vèrtexs del graf estiguin col·locats d'acord amb les coordenades que els corresponen (trobades en la primera pràctica) sobre el mapa de Catalunya. Per fer-ho, caldrà adaptar l'escala utilitzada per a trobar les

coordenades del contorn de Catalunya i de les fronteres entre comarques a l'escala utilitzada en la primera pràctica per trobar les coordenades de les capitals de comarca per tal que coincideixin.

3. **Aplicació**

   Aquest problema és, en definitiva, un problema de coloració de vèrtexs perquè cadascun d'ells representa una comarca que és l'element que es vol pintar.

   (Veure practica/ColorCom/ColorCom.exe del CD)

4. **Funcionament**

   A l'esquerra de la pantalla apareix una llista que relaciona el nombre de cada vèrtex amb la capital de comarca que representa. A la llista també hi figura el nombre del color corresponent a cada comarca. A la dreta de la pantalla apareix el graf obtingut i, un cop s'ha trobat la coloració òptima, sorgeix el mapa polític de Catalunya pintat segons aquesta coloració.

5. **Solució**

   El mapa obtingut mostra la coloració òptima per les comarques.

6. **Observacions**

   - Al clicar amb el botó dret del ratolí s'elaborarà una nova coloració del mapa. De fet, malgrat canviïn els colors, les associacions de comarques d'un mateix color no variaran.
   - Al clicar amb el botó esquerre del ratolí sobre un vèrtex (capital) apareixerà sobre la llista el nom de la comarca que té per capital aquest vèrtex.
   - Es pot observar que falta el vèrtex 34 tant a la llista com en el graf. La raó d'aquesta absència és que aquest vèrtex correspon (d'acord amb la llista elaborada en la primera pràctica) a Terrassa. Cal tenir en compte però, que tant Sabadell com Terrassa són capitals d'una mateixa comarca: el Vallès occidental. És per això que Sabadell i Terrassa s'han unit en un sol vèrtex, el 26 i és per aquesta mateixa raó que el graf utilitzat en aquesta pràctica té un vèrtex menys que el graf utilitzat en la resta de pràctiques.

2.3 Conclusions, valoració i treball futur

Definitivament fer la recerca sobre grafs, problemes on intervenen, personalitats que en el seu moment els varen tractar i algorismes ha estat una tasca divertida, interessant i molt enriquidora. Si ara fa un any, quan tot just havia triat el tema, m'haguessin preguntat: "I doncs? Què n'esperes del teu treball de recerca?" Molt probablement no m'hagués imaginat el creixement que m'ha suposat treballar en aquesta branca de la matemàtica. En primer lloc, com és evident, m'ha permès introduir-me en un petit món del qual, abans de trobar el nom per internet, en sabia ben poc (per no dir gens): la teoria de grafs. I dic "petit" perquè és una branca relativament nova d'aquesta ciència i perquè l'univers de les matemàtiques és molt ampli; però aquesta qualificació no deixa de resultar un xic ridícula sobretot tenint en compte la quantitat d'informació i d'aspectes que per falta de temps i de coneixements no apareixen en el treball. I això sense deixar de pensar en tot allò que encara resulta desconegut i roman amagat esperant ésser descobert i estudiat. Deixant de banda això i seguint amb el que deia, l'estudi d'aquest tema per tal d'elaborar-ne un treball m'ha ajudat a familiaritzar-me amb el llenguatge matemàtic, agafar una peculiar simpatia envers les demostracions (tot i que en el treball no n'apareix cap) i a ampliar horitzons. He gaudit de valent llegint algorismes, demostracions de teoremes i resolucions de problemes. Gairebé tots en un principi m'han resultat incomprensibles però amb paciència, un llapis, un paper, l'ajut del llibre i una mica de tossuderia s'han acabat tornant més clars. La redacció d'algorismes en termes matemàtics tampoc ha estat fàcil; després de llegir-los, arribar a entendre el perquè de cadascun dels seus passos he volgut escriure'ls de nou, a la meva manera, desglossant-los en parts i mirant que cadascuna d'elles s'enllacés correctament amb els passos anteriors i posteriors per tal que tot quadrés. Un altra part rellevant del treball ha estat el disseny de grafs per tal d'exemplificar l'aplicació algorísmica i facilitar la comprensió d'alguns conceptes (com en el cas de les seqüències de De Bruijn). En tots ells, els elements com els costos de les arestes, la distribució d'aquestes, el grau dels vèrtexs... han estat pensats especialment per tal de millorar i completar les explicacions donades i s'ha intentat buscar exemples on apareguessin tots els casos

possibles davant dels quals es poguessin presentar dubtes a l'hora de resoldre'ls (com per exemple la presencia d'arestes formant cicles a l'hora d'aplicar Kruskal o com la presencia de cadascun dels possibles casos exposats en la teoria per tal de reduir dades prèviament a l'aplicació de Sullivan).

De fet, puc confessar que ha estat tant l'interès que en mi ha captat aquest tema i la motivació, que en molts casos m'he permès una escapada per anar més enllà del que era estrictament necessari pel treball llegint llibres sobre matemàtica moderna i sobre la història de les matemàtiques i dels seus descobriments o visitant pàgines web de contingut matemàtic ja sigui enfocat al lleure (curiositats...) o a l'àmbit didàctic.

En segon lloc cal dir que el fet de ser un treball dut a terme al llarg d'un termini de temps relativament llarg ha requerit la dedicació constant per tal de prevenir acumulacions de feina, i corredisses d'última hora. Tot i això no he pogut evitar acabar-lo notant una lleugera sensació de falta de temps acompanyada del Pretèrit Pluscuamperfet del verb poder: "Hagués pogut afegir...", "Hagués pogut fer...", "Hagués pogut canviar..."

Malgrat tot, el treball s'ha hagut d'acabar i he intentat compensar aquest petit sentiment de culpa amb algunes propostes de treball referents al tema que apareixeran més endavant.

En quant a la pràctica, només dir que ha sigut molt laboriosa, principalment pel fet d'anar trobant un a un punts propers entre ells del contorn de Catalunya o de les fronteres entre comarques per tal que el mapa fruit del trasllat de les coordenades al programa emprat fos el més precís possible. Tampoc ha sigut lleugera la feina de comprovar una a una l'existència o no existència dels 861 possibles camins directes corresponents a cadascuna de les possibles combinacions de capitals de comarca. Sobretot, si després de fer-ho s'obté un resultat de 703 camins directes i, per necessitats òbvies d'aclarir el graf resultant, s'han hagut de tornar a fer 703 comprovacions per eliminar-ne les que, malgrat no passar just per una capital de comarca, s'hi apropaven i poder arribar a un nombre d'arestes més raonable. Tot i així, no ha deixat de ser un exercici nou i interessant que m'ha permès aprofundir més en aquests tres apartats concrets de la teoria de grafs.

Per acabar vull presentar alguna conclusió, fer un parell d'observacions i, com he dit anteriorment, obrir noves propostes de treball.

- La funció $f(x) = \frac{x \cdot (x-1)}{2}$ dóna el nombre d'arestes que tindrà un K_x en funció de x.
- Un diagrama de De Bruijn tindrà tantes seqüències de De Bruijn diferents associades com circuits eulerians diferents tingui.
- L'eliminació de les arestes (A, D), (C, D) i (G, E) en la presentació de l'algorisme de Prim s'han argumentat dient que la seva addició a l'arbre comportaria l'aparició de cicles. En la redacció de l'algorisme en termes matemàtics s'expressa que les arestes candidates a entrar en el següent pas a l'arbre han de venir donades per

  C:= (v^*, v_i) tal que $v^* \neq v_i$ essent v^* el vèrtex actual.

  Al addicionar una nova aresta (v^*, v_i') a l'arbre tal que:

  $$(v^*, v_i') = \min_{(v^*, v_i) \in C} \{c(v^*, v_i)\} \text{ on } v^* \in E \text{ i } v_i \notin E$$

  (essent E el conjunt de vèrtexs que formen l'arbre)

  observem que C vé definida per:

  C:= C - (v^*, v_i') + (v_i', v_j) sempre que $v^* \neq v_j$

  D'acord amb això no hi haurà mai cap aresta candidata a entrar properament a l'arbre els vèrtexs situats als extrems de la qual pertanyin ambdós al conjunt de vèrtexs integrants de l'arbre.

  És per aquest motiu que cal aclarir el perquè de l'aparició (cadascuna en el seu moment) de (A, D), (C, D) i (E, G) com a arestes candidates a entrar a l'arbre bo i tenint els vèrtexs dels extrems classificats com a integrants de l'arbre. La raó d'això rau en el fet que la conseqüència que ambdós vèrtexs de l'extrem d'una aresta formin part de l'arbre és l'aparició d'un circuit. Per tant els dos arguments són vàlids a l'hora de seleccionar arestes tot i que és més eficient descartar des d'un bon principi les arestes amb vèrtexs pertanyents a l'arbre en els extrems.
- Una idea que s'havia plantejat en un principi va ser la de trobar quina combinació de carreteres era la corresponent a cada aresta en el graf de les dues primeres pràctiques. De no ser per manca de temps i per resultar una tasca amb una certa complicació (indicar el quilòmetre on

cal canviar, els canvis d'autopista...) hagués estat molt interessant de preparar aquestes pràctiques de manera que clicant sobre cada aresta aparegués el camí que representa.

- També m'hagués agradat poder tractar i adquirir unes nocions bàsiques o aprofundir en els temes de teselacions, aparellaments i planaritat així com fer incís en més problemes la resolució dels quals requerís la utilització de grafs.

Per acabar només dir que els grafs tenen una especial importància en l'àmbit d'enginyeria ja que tenen molta aplicabilitat en temes de xarxes de transport.

2.4 Agraïments

Vull agrair al senyor Antoni Benseny Ardiaca, a en Manel Martínez Pascual i a l'Ana Anton l'ajut que m'han proporcionat i el temps que m'han dedicat.

2.5 Bibliografia i webs

- Basart Muñoz Josep M., *"Grafs: fonaments i algorismes"* , Manuals de la Universitat Autònoma de Barcelona.
- Miguel de Guzman, *Cuentos con Cuentas*, Nivola
- Marcos Constantino, Martínez Jacinto, Hernández Valentín, *Matemáticas de Preuniversitario*, Ediciones S. M.
- www.wiris.com
- www.mat.usach.cl/histmat/html/eule.html
- http://thales.cica.es/rd/Recursos/rd97/Otros/15-1-o-p.htm
- http://almez.pntic.mec.es/~agos0000/Euler.html
- http://www.ctv.es/USERS/pacoga/bella/htm/eulerr.htm
- http://www.matematicas.net/paraiso/juegos.php?id=e_ceuler
- http://www.amejor.com/mates/matematicos/euler.htm#arriba
- http://www.divulgamat.net/weborriak/Historia/HistoriaImagen/Euler.asp
- http://thales.cica.es/rd/Recursos/rd99/ed99-0289-02/biografias/wrhamilton.html
- http://caminantes.metropoliglobal.com/web/biografias/hamilton.htm
- http://www.educared.net/concurso2001/180/wiliam_rowan_hamilton.htm
- http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Kempe.html

- http://www.mat.ucm.es/deptos/am/guzman/cuentosconcuentas/cuatrocolores/cuatrocolores0004.htm
- http://centros5.pntic.mec.es/ies.ortega.y.rubio/Mathis/Colores/4colores.htm
- http://thales.cica.es/rd/Recursos/rd99/ed99-0289-02/biografias/acayley.html
- http://bacterio.uc3m.es/docencia/profesores/antonio/biog/kirchhoff.htm

3. Annex

3.1 Vocabulari

- Arborescència 34
- Arbre 30
  - de Steiner 32
  - generador 30
  - òptim 32
- Arc 16
- Aresta 16
- Arrel 34
- Articulació 25
- Berge 10
- BFS 31
- Camí 17
  - elemental 17
  - eulerià 35
  - hamiltonià 40
  - simple 17
- Cayley 7
- Circuit 17
  - eulerià 35
  - hamiltonià 40
- Coloració
  - d'arestes 46
  - d'un mapa 9
  - de vèrtexs 45
- Component 24
- CPP 39
- De Bruijn 36
  - diagrama 37
  - seqüència 36
- De Morgan 9
- DFS 31
- Dijkstra 32
- Distància 18
- Dodecàedre 44
- Edmonds 32
- Erdös 10
- Euler 6
- Fleury 35
- Floyd 32
- Ford 32
- Goldberg 10
- Graf
  - complementari 22
  - complet 22
  - dirigit 12
  - eulerià 50
  - hamiltonià 40
  - k-regular 15
  - mixt 13
  - mixt balancejat 20
  - mixt parell 21
  - parcial 18
  - pla 43
  - r-colorable 45
  - simètric 12
  - simètric subjacent 23

trivial 18
- Grau 14
  exterior 14
  interior 14
- Guthrie 9
- Haken 9
- Hamilton 7
- Harary 10
- Heawood 9
- Hectàedre 44
- Icosàedre 44
- Kempe 8
- Kirchhoff 10
- Königsberg 47
- Kruskal 32
- Kuratowsky 10
- Lovász 10
- Llaç 16
- Llista d'adjacència 29
- Matriu
  d'adjacència 28
  de graus entrant 30
- Moore 32
- Multigraf 21
- Multiplicació llatina 41
- Nombre cromàtic 45
- Octàedre 44
- Políedre 44
- Pont 24
- Prim 32
- Problema
  del carter xinès 39
  del laberint 53
  del viatjant 42
- Regió 43
- Roberts 41
- Separador 26
- Subgraf 19
  maximal 20
  minimal 20
- Sullivan 32
- Tall 27
- Tarjan 10
- Tetràedre 44
- TSP 42
- Tutte 10
- Vèrtex 10
  adjacent 14
  antecessor 13
  successor 13
  veí 14
- Withney 10

3.2 Camps vinculats

3.2.1 Algorismes

*"Un **algorisme** és un conjunt finit d'instruccions o passos que serveixen per a executar una tasca o resoldre un problema (...). D'una manera més formal, un algorisme és una seqüència finita d'instruccions realitzables, no ambigües, l'execució de les quals condueix a una resolució d'un problema."*

http://ca.wikipedia.org/wiki/Algorisme

En la teoria de grafs intervenen nombrosos algorismes. A continuació se'n detallen uns quants dels que han aparegut al llarg del treball.

- Algorisme **BFS**: aquest algorisme s'utilitza per trobar l'arbre BFS.

  1. *Etiquetar v_1 amb un 0.*
  2. *j:=0.*
  3. *Buscar tots els vèrtexs adjacents a un vèrtex j no etiquetats.*
     Si no n'hi ha s'ha acabat.
     Si n'hi ha anar a 4.
  4. *Etiquetar tots els vèrtexs trobats en 3 amb j+1.*
  5. *j:=j+1*
     Tornar a 3.

(Veure BFS del CD)

- Algorisme **DFS**: és l'algorisme utilitzat a l'hora de trobar un arbre DFS.

  1. *Considerar inicialment totes les connexions aresta-vèrtex no etiquetades.*
     $v:=v_1$
  2. *Si totes les connexions a v són etiquetades anar a 4.*
     Si no, anar a 3

3. *Agafar una connexió no etiquetada, etiquetar-la amb un 0 i recórrer l'aresta fins arribar a l'extrem v_i.*
   Si v_i té cap connexió etiquetada, etiquetar la connexió d'entrada a v_i amb un 0 i tornar enrere cap a v. Tornar a 2.
   Si no, etiquetar la connexió d'entrada a v_i amb un 1, v:=v_i. Tornar a 2.
4. *Si hi ha cap connexió etiquetada amb un 1, usar-la per anar a l'altre extrem de l'aresta v_i. v:=v_i. Tornar a 2.*
   Si no, s'ha acabat.

(Veure DFS del CD)

- Algorisme de Sullivan: aquest algorisme s'utilitza per a trobar un arbre de cost mínim aproximat a l'arbre de Steiner òptim.

  Sigui: V_1 el conjunt de vèrtexs que cal unir.

1. *Elaborar el $K_z = G_1(V_1, A_1)$ a partir de G(V, A) on z = $/Z/$ i Z = V_1. Cada c_{ij} en $K_z = c(C_{ij})$ en G.*
2. *Buscar per Kruskal o Prim l'arbre de cost mínim T_1 per K_z.*
3. *Substituir cada aresta (v_i, v_j) en K_z per les corresponents arestes que formen C_{ij} en G. Aquest nou graf serà G_2.*
   Recuperar el valor inicial del cost de cada aresta en G_2.
4. *Buscar l'arbre de cost mínim T_2 per Kruskal o Prim de G_2.*
   T_2 és l'arbre de cost mínim aproximat a l'arbre de Steiner per a Z.

(Veure Sullivan del CD)

- Algorisme de Moore: Algorisme utilitzat a l'hora de buscar un arbre òptim d'un graf simètric amb un mateix cost en cada aresta.

Sigui: w el vèrtex inicial

$\lambda(v_i)$ l'etiqueta amb el cost que s'ha arrossegat des de w fins a v_i

f el vèrtex final

$\lambda(f) = c(C_{wf})$

v* el vèrtex actual

1. *Inicialment:* $v^* := w$

   $\lambda(v^*) := 0$

   $i := 0$

2. *Buscar tots els vèrtexs* $v_j \in \Gamma(v^*)$.

   Si $\Gamma(v^*) = \varnothing$, *s'ha acabat.*

   Si no, etiquetar v_j *amb* $\lambda(v_j) := i + 1$ *i anar a 3.*

3. *Actualitzar:* $v^* := v_j$

   $i := i + 1$

   Si $f = v^*$, *s'ha acabat.*

   Si no, anar a 2.

- Algorisme de Dijkstra: Algorisme emprat a l'hora de trobar l'arbre de cost mínim en grafs dirigits amb costos no negatius en els arcs. No obstant, també es pot aplicar a grafs simètrics.

Sigui: w el vèrtex inicial

f el vèrtex final

L és el conjunt de vèrtexs recorreguts

V és el conjunt

1. *Inicialment:* $\lambda(w) := 0$

   $L := \{w\}$

   $V' := V - L$

2. *Buscar* $v_i \in \Gamma(L)$, $v_i \in V'$ *tal que* $\lambda(v_i)$ *sigui mínim, essent* $\lambda(v_i) = \lambda(v_{i-1}) + c(v_{i-1}, v_i)$ *i essent* $v_{i-1} \in L$ *i* $v_{i-1} \in \Gamma^{-1}(v_i)$.

3. *Si V' = ∅, s'ha acabat i no s'ha trobat f.*

   Si no, actualitzar: $v_j := v_i$

   Anar a 4.

4. *Si f = v_j, s'ha acabat i $\lambda(v_j) = c(w,f)$.*

   Si no, actualitzar: $L := L + \{v_j\}$

   Tornar a 2.

(Veure Dijkstra del CD)

- Algorisme de Floyd:

  Sigui: $\lambda_{ij}^{m} = c(C_{ij})$

  $V = \{ v_1, v_2, \ldots, v_n \}$

  m = pas

  $\{ v_i, v_j \} \in V$

1. *Inicialment: m:= 1*

$$\lambda_{ii}^{0} = 0$$

$$\lambda_{ii}^{0} = \begin{cases} c(v_i, v_j) & si\ (v_i, v_j) \in A \\ \infty & si\ (v_i, v_j) \notin A \end{cases}$$

2. *Buscar per tot arc (v_i, v_j):*

$$\lambda_{ij}^{m} = min\ \{ \lambda_{ij}^{m-1}, \lambda_{im}^{m-1} + \lambda_{mj}^{m-1} \}$$

3. *Si m = n, s'ha acabat.*

   Si no, anar a 4.

4. *Actualitzar:* $m := m + 1$

   Tornar a 2.

- Algorisme de Fleury: algorisme utilitzat a l'hora de buscar circuits eulerians.

Sigui: L la llista que conté ordenats tots els vèrtexs pels quals s'ha anat passant
v* el vèrtex actual
A' el conjunt d'arestes recorregudes
$A(v_i)$ la llista de vèrtexs adjacents a v_i dins el graf G–A'
w el vèrtex inicial

1. *Es parteix de manera que es té:* $L:=w$
   $v^*:=w$
   $A':=\varnothing$
2. *Si* $|A(v^*)|>0$ *anar a 3*
   Si no, s'ha acabat. Anar a 4.
3. *Si* $|A(v^*)|>1$ *buscar un vèrtex* $v\in A(v^*)$ *tal que* (v^*, v) *no sigui un pont en* $G-A'$
   Si no, v és l'únic vèrtex de $A(v^*)$.
   Actualitzar: $A(v^*):= A(v^*)-\{v\}$
   $A(v):= A(v)-\{v^*\}$
   $A':= A'\cup\{(v^*, v)\}$
   $v^*:=v$
   $L:= L\cup v^*$
   Tornar a 2.
4. *El resultat és L*

(Veure Fleury del CD)

Si es pot realitzar aquest algorisme en un graf G de manera que en el pas 3 sempre sigui possible escollir la instrucció "Si", la L que en resultarà contindrà un circuit eulerià de G.

- Algorisme emprat en el circuit eulerià mixt:

1. *V són vèrtexs nous*
   Escollir un vèrtex inicial $v^*\in V$; $v:=v^*$

2. *Si v és nou, etiquetar-lo com a usat i estendre l'arborescència a partir de v. Anar a 3.*
   Si no, explorar-lo. Anar a 4.
3. *Si v=v* i l'arborescència és màxima s'ha acabat.*
   Si no, anar a 2.
4. *Si es troba un vèrtex nou v_j tal que $\exists (v_j, v)$, $v:=v_j$. Tornar a 2.*
   Si no, s'etiqueta v com a vèrtex explorat. Anar a 5.
5. *Assignar a (vi, v_j) el sentit de v_i a v_j, éssent (v_i, v_j) una aresta tal que $v_j=v$; $v:=v_i$.*
   Si v=v i l'arborescència és màxima, s'ha acabat*
   Si no, tornar a 2 o 4 (4 si sempre v és explorat).

- Algorisme d'Edmonds: aquest algorisme s'utilitza en el problema del carter xinès per a grafs no eulerians simètrics. Es basa en la duplicació d'algunes de les arestes per tal de convertir el graf en eulerià. Aquesta duplicació però, s'ha de fer tenint en compte el cost de cada aresta ja que aquest ajudarà a determinar la repetició d'una o d'una altra. En resum, aquest mètode controla el fet d'una obtenció d'un circuit eulerià mitjançant la duplicació d'arestes en base que el cost addicional afegit sigui mínim.

Sigui: G(V,A) el graf no eulerià simètric
C_{ij} el camí de cost mínim
$C(C_{ij})$ el cost de C_{ij}
K_S el graf complet obtingut a partir de $V_S \in V$

1. *Dividir V en dos subconjunts:* $V_P := \{v_i \in V$ *tal que grau* (v_i) *és parell}*
   $V_S := \{v_i \in V$ *tal que grau* (v_i) *és senar}*
   $s := |V_S|$
2. *Calcular* $c(C_{ij})$ *tal que* $v_i \in V_S$, $v_j \in V_S$ *i* $v_i \neq v_j$ *de totes les parelles entre els vèrtexs de* V_S
3. *Elaborar* K_S *associant els* $c(C_{ij})$ *trobats en 2 a les corresponents arestes* (v_i, v_j).

4. *Cercar un aparellament perfecte òptim M* en K_S.*
5. *Duplicar en G les arestes que formaven els C_{ij} corresponents als costos de les arestes de M**
6. *Resulta un multigraf eulerià.*

(Veure Edmonds del CD)

- Algorisme de Kruskal: Aquest algorisme s'utilitza en la recerca d'arbres de cost mínim. Consisteix en crear un arbre d'un graf G a partir de diversos components que van sorgint a partir de les arestes de costos mínims. Al final s'obtindrà un graf fruit de la unió de tots les components.

1. *Sigui: G(V, A) el graf del qual es vol trobar l'arbre de cost mínim*
   $G_0(V_0, A_0=\varnothing)$ el graf que serà l'arbre de cost mínim de G
   k:=0
2. *Elaborar una llista de les arestes de G ordenades de menor a major segons el cost*
3. *Prendre la primera aresta $a_i = (v_i, v_j)$ no usada de la llista que no formi circuit a G_k.*
   Actualitzar: k:=k+1
   $G_k := G_{k-1} \cup \{a_{ij}\}$
   $V_k := V_{k-1} \cup \{v_i, v_j\}$ sempre que v_i o $v_j \notin V_{k-1}$
   $A_k := A_{k-1} \cup \{a_{ij}\}$
4. *Si $k < n - 1$, essent $n = |V|$; tornar a 3.*
   Si no, acabar.

(Veure Kruskal del CD)

- Algorisme de Prim: Aquest algorisme també s'utilitza, com el de Kruskal, per a buscar arbres de cost mínim però, a diferència d'aquest, l'arbre creix estenent un sol component. L'arbre es forma com a resultat d'anar addicionant cada cop l'aresta menys costosa possible al component. Finalment s'obté el mateix resultat pels dos procediments (Kruskal i Prim) a menys que l'arbre de cost mínim del graf en qüestió no sigui

únic. Si es dóna aquesta situació els dos algorismes poden donar com a resultat dos arbres diferents tot i que d'igual cost.

Sigui: E: {vèrtexs de l'arbre que es va formant}
R: {arestes de l'arbre que es va formant}
C: {arestes candidates a entrar properament a l'arbre}
v*: vèrtex actual

1. $E := \{w\}$
   $R := \varnothing$
   $v^* := w$
   $C := (v^*, v_i)$ tal que $v^* \neq v_i$
2. Agafar (v^*, v_i') tal que $(v^*, v_i') = \min_{(v^*, v_i) \in C} \{c(v^*, v_i)\}$ on $v^* \in E$ i $v_i \notin E$
3. Actualitzar: $E := E \cup \{v_i'\}$
   $R := R \cup \{(v^*, v_i')\}$
   $C := C - (v^*, v_i') + (v_i', v_j)$ sempre que $v^* \neq v_j$
   $v^* := \{v^*, v_i'\}$
   $v_j := v_i$
4. Si C=0, acabar.
   Si no, tornar a 2.

(Veure Prim del Cd)

3.2.2 Programació

El llenguatge de programació utilitzat en la part pràctica ha estat el C++ i l'entorn de programació emprat ha estat el del Visual Studio 6.0.

Consta d'una primera part on es recullen totes les dades necessàries per a l'execució del graf relatives a les diferents parts que el componen:

Als vèrtexs se'ls assigna una identificació, una etiqueta...

A continuació s'introdueixen les dades referents a les arestes i la resta d'informació necessària per l'execució del graf: l'adjacència de determinats vèrtexs, les coordenades dels punts que constituiran la imatge, l'escala desitjada per mostrar aquesta imatge...