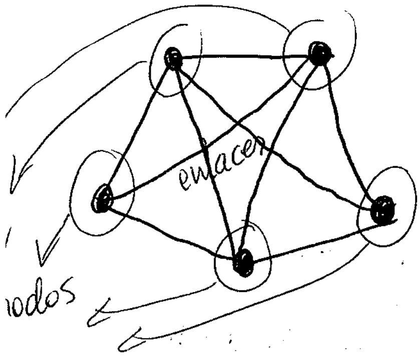
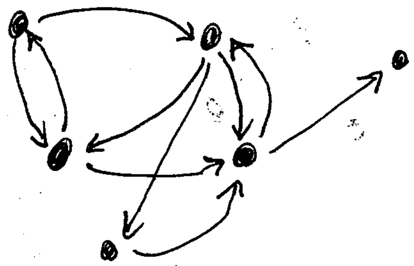
INTRODUCCIÓN GRAFOS



GRAFO DE INTERNET

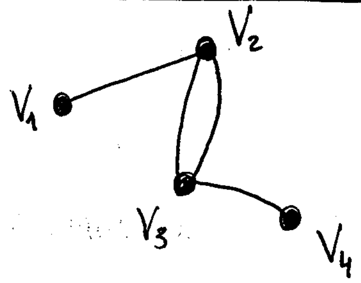


GRAFO $G = \{E, V\}$

E: aristas o enlaces

V: nodos o vértices

GRAFO NO DIRIGIDO



$G \equiv \{E, V\}$

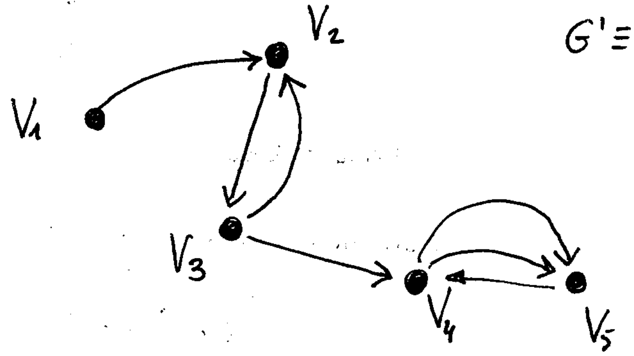
$V \equiv \{V_1, V_2, V_3, V_4\}$

$E \equiv \{\underbrace{\{V_1, V_2\}}, \{V_2, V_3\}^{(I)}, \{V_2, V_3\}^{(II)}, \{V_3, V_4\}\}$

enlace = par no ordenado

$\{V_1, V_2\} \equiv \{V_2, V_1\}$

GRAFO DIRIGIDO



$G' \equiv \{E', V'\}$

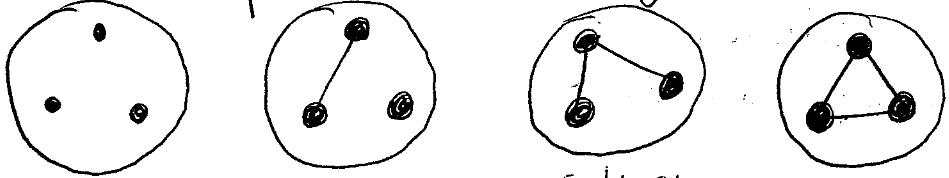
$V' \equiv \{V_1, V_2, V_3, V_4, V_5\}$

$E' \equiv \{\underbrace{(V_1, V_2)}, (V_2, V_3), (V_3, V_2), (V_3, V_4), (V_4, V_5)^{(I)}, (V_4, V_5)^{(II)}, (V_5, V_4)\}$

enlace = par ordenado (vértice origen, vértice destino)

GRAFO SIMPLE (ND)

Entre cada par de vértices hay como máximo UNA ARISTA.



GRAFO ND SIMPLE con n vértices.

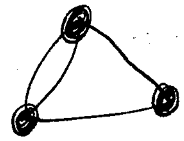
¿cuál es el nº máximo de enlaces?

enlace $\{\underset{n}{\cdot}, \underset{n-1}{\cdot}\}$

$$\binom{n}{2} = \frac{n!}{2!(n-2)!} = \frac{n(n-1)}{2}$$

MULTI GRAFO

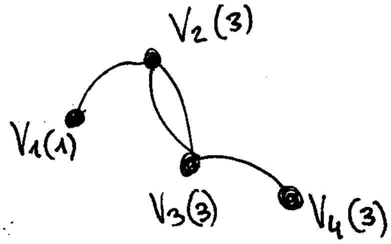
Entre al menos un par de vértices hay aristas paralelas



TEOREMA N.D.

$$\boxed{\sum_{v \in V} \delta(v) = 2|E|}$$

$\delta(v)$ = GRADO del VÉRTICE V $\equiv$ nº de enlaces incidentes



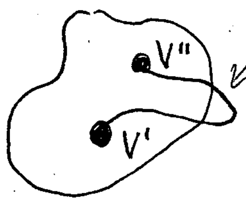
#enlaces = 5

$$\underset{\delta(V_1)}{1} + \underset{\delta(V_2)}{3} + \underset{\delta(V_3)}{3} + \underset{\delta(V_4)}{3} = 2.5$$

Demostración

Suponemos que en un grafo ($G = \{E, V\}$) con $|E|$ (#enlaces de E) enlaces se cumple:

$$\sum_{v \in V} \delta^G(v) = 2|E|$$



enlace $|E|+1$

$\delta^{G'}(v') \equiv \delta^G(v') + 1$ = grado de v' en el nuevo grafo.

$\delta^{G'}(v'') \equiv \delta^G(v'') + 1$ = grado de v'' en el nuevo grafo

$$\underset{\substack{\| \\ 2|E|}}{\sum_{v \in V} \delta^{G'}(v)} = \underset{\substack{\| \\ 2(|E|+1)}}{\sum_{v \in V} \delta^G(v) + 2}$$

$|E|+1$ → nº de enlaces de G'

Caso base inducción:

Grafo con $|E| = 0$

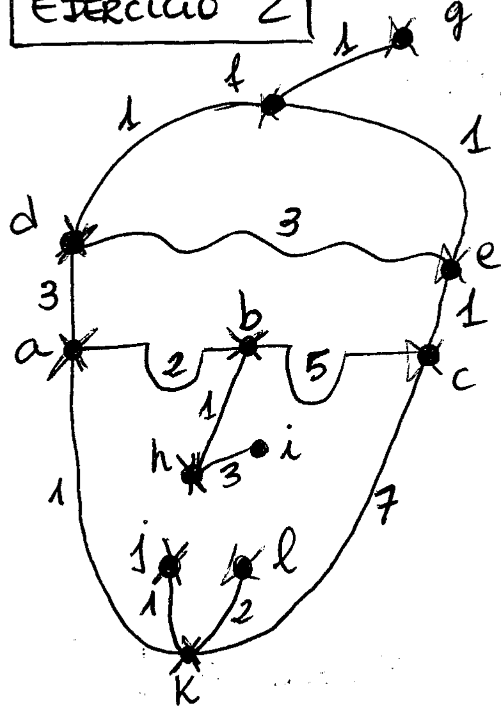
V_1 (0)

$0 = 2.0$

grado (V_1) → 0

$|E|$ = nº enlaces

EJERCICIO 2



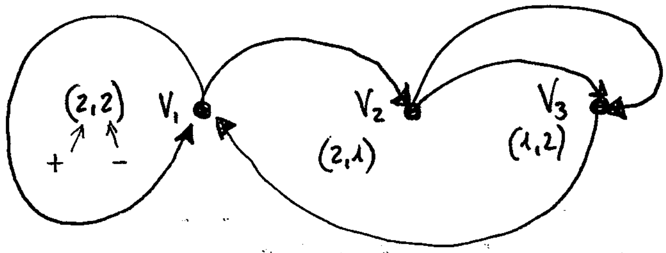
$c \leftarrow e \leftarrow f \leftarrow d \leftarrow a$

	L(∅)	L(1)	L(2)	L(3)	L(4)	L(5)	L(6)	L(7)	L(8)	L(9)	L(10)
a	(∅)*	–	–	–	–	–	–	–	–	–	–
b	∞	2a	(2a)*	–	–	–	–	–	–	–	–
c	∞	∞	8k	7b 7b<8k	7b	7b	7b	7b	7b	6e 6e<7b	(6e)*
d	∞	3a	3a	3a	(3a)*	–	–	–	–	–	–
e	∞	∞	∞	∞	∞	6d	6d	6d	(5f)* 6d>5f	–	–
f	∞	∞	∞	∞	∞	4d	4d	(4d)*	–	–	–
g	∞	∞	∞	∞	∞	∞	∞	∞	5f	(5f)*	–
h	∞	∞	∞	3b	3b	(3b)*	–	–	–	–	–
i	∞	∞	∞	∞	∞	∞	6h	6h	6h	6h	6h
j	∞	∞	2k	(2k)*	–	–	–	–	–	–	–
k	∞	(1a)*	–	–	–	–	–	–	–	–	–
l	∞	∞	3k	3k	3k	3k	(3k)*	–	–	–	–

TEOREMA
GRAFO
DIRIGIDO

$$\boxed{\sum_{v \in V} \delta^{(+)}(v) + \sum_{v \in V} \delta^{(-)}(v) = 2 \cdot |E|}$$

(−) = entrantes incidentes



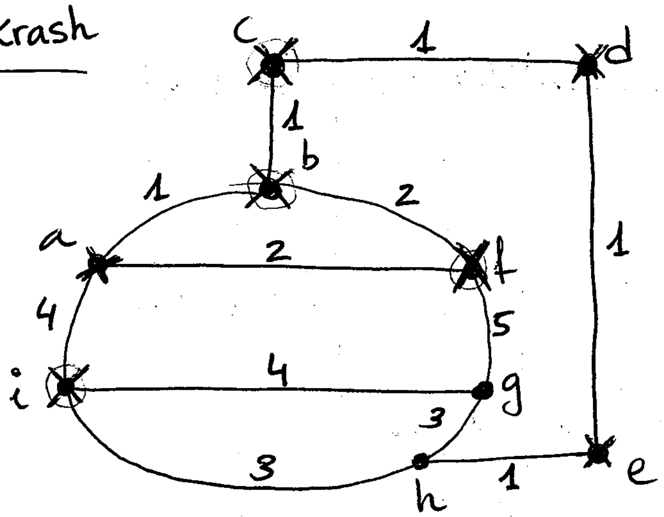
$|E| = 5$

$$\underbrace{\sum_{v \in V} \delta^{(+)}(v)}_{2+2+1} + \underbrace{\sum_{v \in V} \delta^{(-)}(v)}_{2+1+2} = \underbrace{2|E|}_{2.5}$$ ✓

EJERCICIO

Kandy-Krash

$A \rightarrow G$

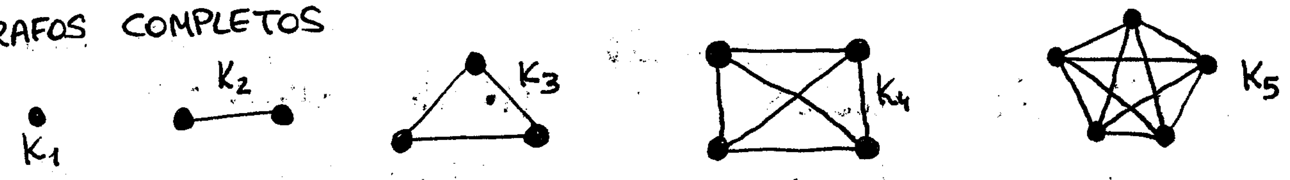


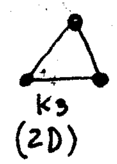
	$L(\emptyset)$	$L(1)$	$L(2)$	$L(3)$	$L(4)$	$L(5)$	$L(6)$	$L(7)$	$L(8)$
a	$(\emptyset)^*$	—	—	—	—	—	—	—	—
b	∞	$(1_a)^*$	—	—	—	—	—	—	—
c	∞	∞	$(2_b)^*$	—	—	—	—	—	—
d	∞	∞	∞	3_c	$(3_c)^*$	—	—	—	—
e	∞	∞	∞	∞	∞	$(4_d)^*$	—	—	—
f	∞	2_a	2_a $3_b > 2_a$	$(2_a)^*$	—	—	—	—	—
g	∞	∞	∞	∞	7_f	7_f	7_f	7_f $8_i > 7_f$	$(7_f)^*$ $8_h > 7_f$
h	∞	∞	∞	∞	∞	∞	5_e	$(5_e)^*$ $7_i > 5_e$	—
i	∞	4_a	4_a	4_a	4_a	4_a	$(4_a)^*$	—	—

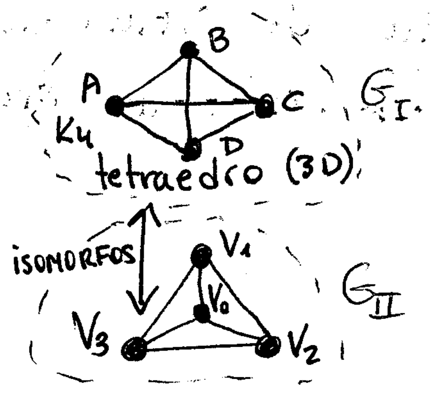
$\boxed{g \leftarrow f \leftarrow a}$

GRAFOS

- GRAFOS COMPLETOS



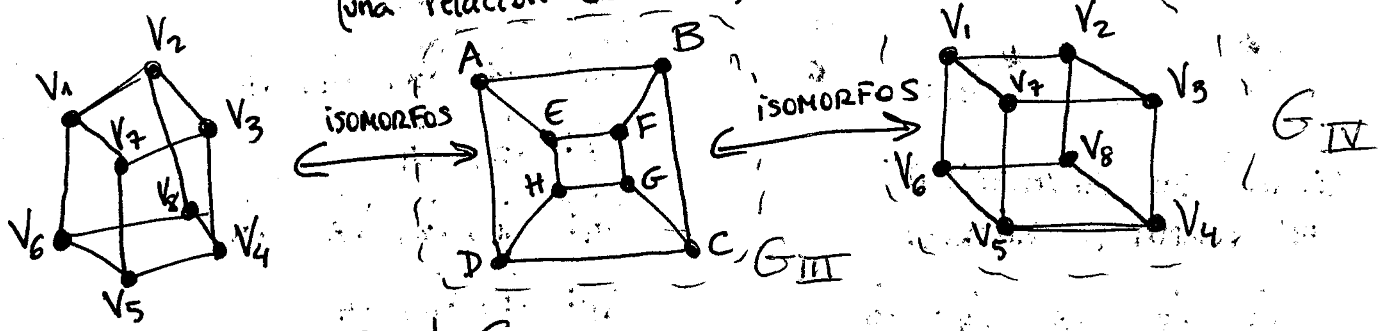




K5 SIMPLEX
(4D)

Vert. G_I	Vert. G_{II}
A ⟷	V_3
B ⟷	V_1
C ⟷	V_2
D ⟷	V_0

(una relación de varias)



G_{III}	G_{IV}
A ⟷	V_6
B ⟷	V_8
C ⟷	V_4
D ⟷	V_5
E ⟷	V_1
F ⟷	V_2
G ⟷	V_3
H ⟷	V_7

PERMUTACIONES DE NODOS

a b c

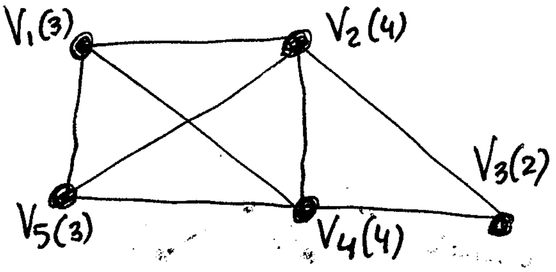
perm. de orden 1:
- b a c (1° ⟷ 2°)
- c b a (1° ⟷ 3°)
- a c b (2° ⟷ 3°)

perm. de orden 2:
- c a b
- b c a

INVARIANTES bajo un ISOMORFISMO

- Nº de nodos (dificultad 0) y n de enlaces (dificultad 1)
- Estructa de adyacencia ("dificultad completa")
  ↳ lista de grados de nodos (dificultad 2)

REPRESENTACIÓN DE GRAFOS



LISTA DE ADYACENCIA

- V_1: V_2, V_4, V_5
- V_2: V_1, V_3, V_4, V_5
- V_3: V_2, V_4
- V_4: V_1, V_2, V_3, V_5
- V_5: V_1, V_2, V_4

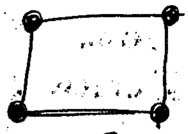
LISTA DE ADYACENCIA EN FORMA DE MATRIZ:

$$
\begin{array}{c}
 \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5
\end{array}
\begin{array}{c}
\begin{array}{ccccc} V_1 & V_2 & V_3 & V_4 & V_5 \end{array} \\
\begin{pmatrix}
0 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 0
\end{pmatrix}
\end{array}
\begin{array}{l}
 \\ \rightarrow 3 \\ \rightarrow 4 \\ \rightarrow 2 \\ \rightarrow 4 \\ \rightarrow 3
\end{array}
$$

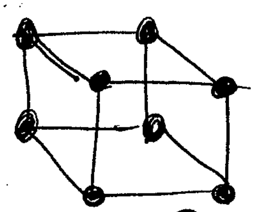
→ la suma = grado del vértice de la fila

D=0

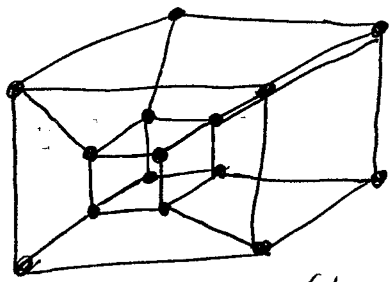
D=1



D=2

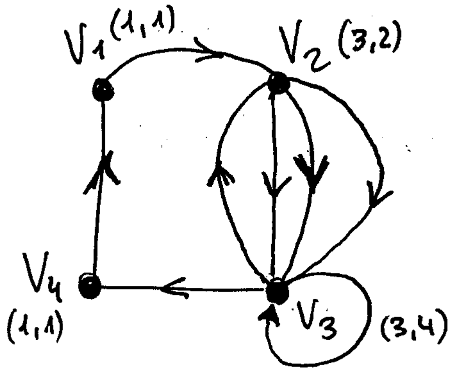


D=3



D=4 (teseracto)

CODIFICACIÓN DE GRAFOS DIRIGIDOS

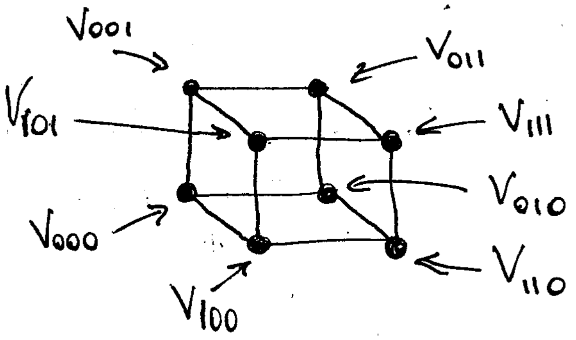


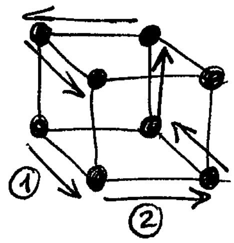
$$
\begin{array}{c|cccc|c}
 & V_1 & V_2 & V_3 & V_4 & \\
\hline
V_1 & 0 & 1 & 0 & 0 & \to 1 \\
V_2 & 0 & 0 & 2 & 0 & \to 3 \\
V_3 & 0 & 1 & 1 & 1 & \to 3 \\
V_4 & 1 & 0 & 0 & 0 & \to 1 \\
\hline
 & \downarrow 1 & \downarrow 2 & \downarrow 4 & \downarrow 1 &
\end{array}
$$

salientes (filas); $\underbrace{1 \quad 2 \quad 4 \quad 1}_{\text{entrantes}}$

TRAYECTORIA

Trayectoria: Secuencia alternada de vértices y enlaces, que comienza y termina con 1.





$V_{000} \to V_{100} \to V_{110} \to V_{010} \to V_{010} \to V_{011} \to V_{001} \to V_{101} \to V_{111}$

solo cambia un bit entre vértices consecutivos

distancia de Hamming (entre palabras formadas por el mismo nº de bits) = nº de bits en los que se diferencian las palabras

$\boxed{\text{CÓDIGOS DE GREY}}$

Trayectoria simple: No hay aristas repetidas (puede haber vértices repetidos)

Ciclo o circuito: Trayectoria (de long>0) que comienza y termina en el mismo vértice.

Circuito simple: No hay aristas repetidas.

CONECTIVIDAD EN GRAFOS

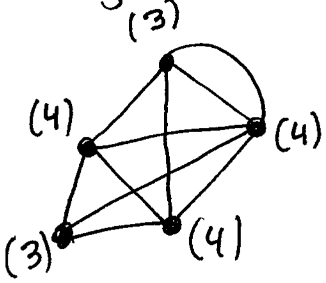
Grafo (no dirigido) conexo: Existe una trayectoria entre cualquiera 2 vértices del grafo.

Grafo no conexo: Unión de dos o más grafos conexos que no comparten vértices.

Subgrafos del grafo no conexo "COMPONENTES CONEXAS"

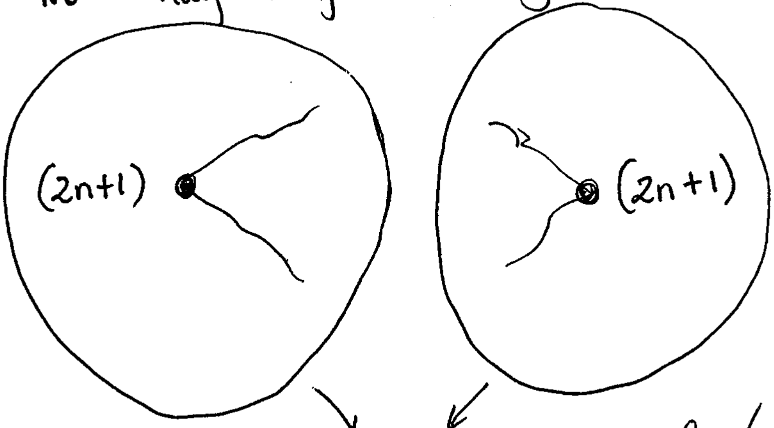
TEOREMA

Si un grafo tiene exactamente 2 vértices de grado impar hay al menos una trayectoria que une dichos vértices.



Demostración por reducción al absurdo

Suponemos que existe un grafo con exactamente 2 vértices de grafo impar, pero para el que no hay ninguna trayectoria que los une.



2 componentes del grafo (grafo no conexo)
[no comparten ningún vértice.]

$$\sum_{v \in V} \delta(v) = 2|E|$$

Por consiguiente, 2 subgrafos no conectados; cada uno ~~con~~ de los cuales tiene exactamente 1 nodo de grafo impar $\Rightarrow$

$\Rightarrow$ la suma de grados sería impar. $\longleftrightarrow$

CONTRADICCIÓN con $\sum_{v \in V} \delta(v) = \boxed{2|E|}$

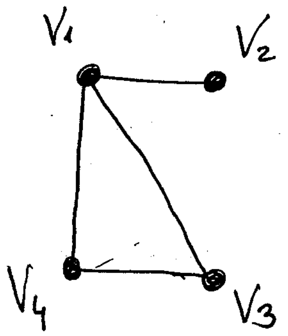
TEOREMA

El número máximo de aristas de un grafo simple no conexo con N vértices y K componentes es:

$$\binom{n}{2} = \frac{n!}{2!(n-2)!} = \frac{n(n-1)}{2} \quad ?$$

INVARIANTES ENTRE GRAFOS ISOMORFOS

* Igual número de vértices (o nodos).
* Igual número de aristas (o enlaces).
* Lista de grados de los vértices.
* Nº de circuitos con longitud K.



MATRIZ DE ADYACENCIA

$$A = \begin{array}{c|cccc} & V_1 & V_2 & V_3 & V_4 \\ \hline V_1 & & & & \\ V_2 & & & & \\ V_3 & & & & \\ V_4 & & & & \end{array} \quad A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

$$A^4 = \begin{pmatrix} 11 & 2 & 6 & 6 \\ 2 & 3 & 4 & \boxed{4} \\ 6 & 4 & 7 & 6 \\ 6 & 4 & 6 & 7 \end{pmatrix}$$

Hay 4 trayectorias distintas de longitud 4 entre $V_2 \rightarrow V_4$

$V_2 \rightarrow V_1 \rightarrow V_3 \rightarrow V_1 \rightarrow V_4$

$V_2 \rightarrow V_1 \rightarrow V_2 \rightarrow V_1 \rightarrow V_4$

$V_2 \rightarrow V_1 \rightarrow V_4 \rightarrow V_3 \rightarrow V_1$

$V_2 \rightarrow V_1 \rightarrow V_4 \rightarrow V_1 \rightarrow V_4$

TEOREMA

El nº de trayectorias de longitud r entre V_i y V_j, que son vértices del grafo G, cuya matriz de adyacencias es A, es $\cancel{[A^r]_{ij}}$ el elemento $[A^r]_{ij}$

MULTIPLICACIÓN DE MATRICES

$$D = B.C$$

$$D_{ij} = \sum_{k=1}^{N} B_{ik} C_{kj}$$

Demostración

$$[A^r]_{ij} = \sum_{k=1}^{M} [A^{r-1}]_{ik} A_{kj}$$

$$A^4 = A^3 . A \longrightarrow \underbrace{(A^4)_{ij}}_{} = \sum_{k=1}^{M} \underbrace{(A^3)_{ik}}_{} A_{kj}$$

→ nº de trayectorias de longitud 4 entre V_i y V_j.

→ nº de trayectorias de longitud 3 entre V_i y V_k

→ nº de trayectorias de longitud 1 entre V_k y V_j

$\parallel\!\!\parallel$

nº enlaces $V_k \rightarrow V_j$

$(A^3)_{ik} \cdot A_{kj} \equiv$ se forman $V_i \xrightarrow{(3)} V_k \xrightarrow{(1)} V_j$

TRAYECTORIA EULERIANA: Trayectoria que incluye cada → "todas y cada una" arista exactamente una vez.

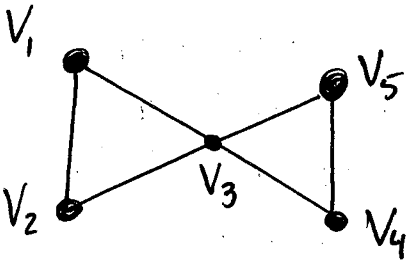
CIRCUITO EULERIANO: Circuito en el que cada arista es incluida exactamente una vez.

GRAFO EULERIANO: Grafo que tiene ~~al menos~~ un circuito Euleriano.

TRAYECTORIA HAMILTONIANA: Incluye cada → "todos y cada uno" vértice exactamente una vez.

CIRCUITO HAMILTONIANO:

GRAFO HAMILTONIANO: Grafo que tiene ~~al menos~~ un circuito hamilton



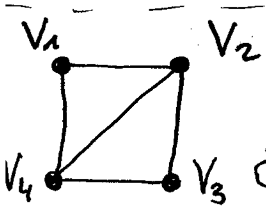
$V_1 \to V_3 \to V_4 \to V_5 \to V_3 \to V_1 \to V_2$ (trayect. euleriana)

$V_2 \to V_3 \to V_1 \to V_2 // V_3 \to V_1 \to V_2 \to V_3$ (circuito longitud =

$V_1 \to V_2 \to V_3 \to V_4 \to V_5 \to V_3 \to V_1$ (circuito euleriano)

$V_1 \to V_2 \to V_3 \to V_4 \to V_5$ (trayectoria hamiltoniana)

Sin circuito hamiltoniano.



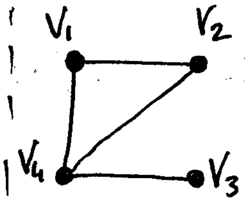
GRAFO HAMILTONIANO

$V_1 \to V_2 \to V_3 \to V_4$

CIRCUITO HAMILTONIANO

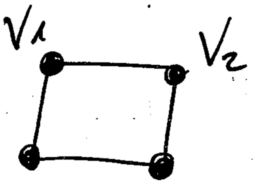
$V_1 \to V_4 \to V_2 \to V_3$

TRAYECTORIA HAMILTONIANA



GRAFO NO HAMILTONIANO

GRAFO EULERIANO



TEOREMA

Un grafo es Euleriano si y solo si todos y cada uno de los vértices tiene grado par.

TEOREMA

que no sea circuito

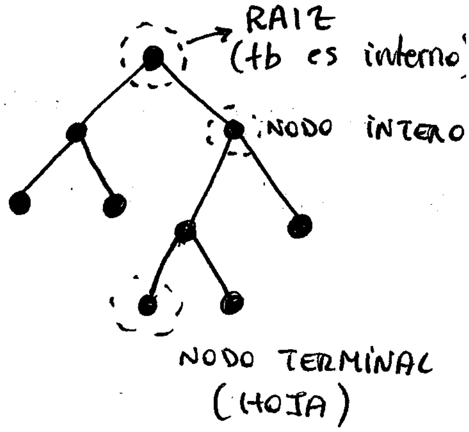
euleriana

Un grafo tiene una trayectoria ~~hamiltoniana~~ si y solo si tiene exactamente dos nodos de grado impar.

Árboles →

ÁRBOLES

Un ÁRBOL es un grafo conexo no dirigido sin circuito simples.



n_1 es ANTECESOR de n_2
- → n_1 es padre de n_2
- o (exclusivo)
- → n_1 es antecesor del padre de n_2

spanning = abarcador

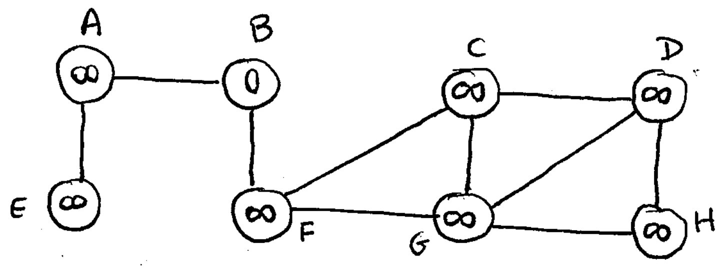
ALGORITMOS

CÓDIGO DE COLORES
- BLANCO: Nodo no visitado
- GRIS: Nodo visitado (o descubierto) pero no expandido
- NEGRO: Nodo visitado y expandido.

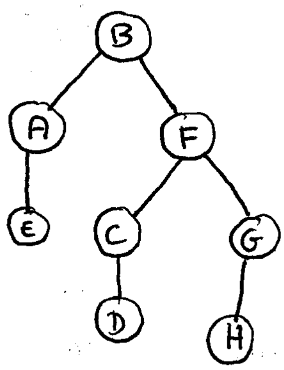
Búsqueda en anchura

Q → FIFO QUEUE (first in, first out)

→ lista de nodos descubiertos, pero aún <u>no explorados</u> ordenada de forma que los primeros de la lista son los nodos descubiertos antes



BREADTH-FIRS TREE



$Q = \{B_0\}$

$Q = \{A_1, F_1\}$

$Q = \{F_1, E_2\}$

$Q = \{E_2, C_2, G_2\}$

$Q = \{C_2, G_2\}$

$Q = \{G_2, D_3\}$

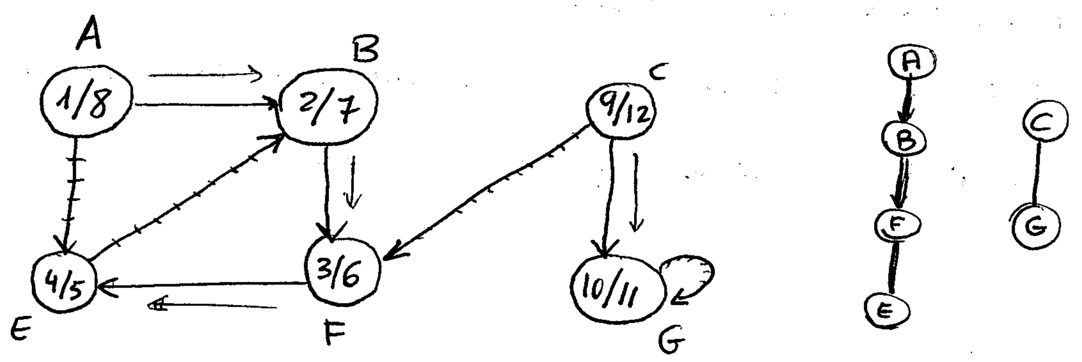
$Q = \{D_3, H_3\}$

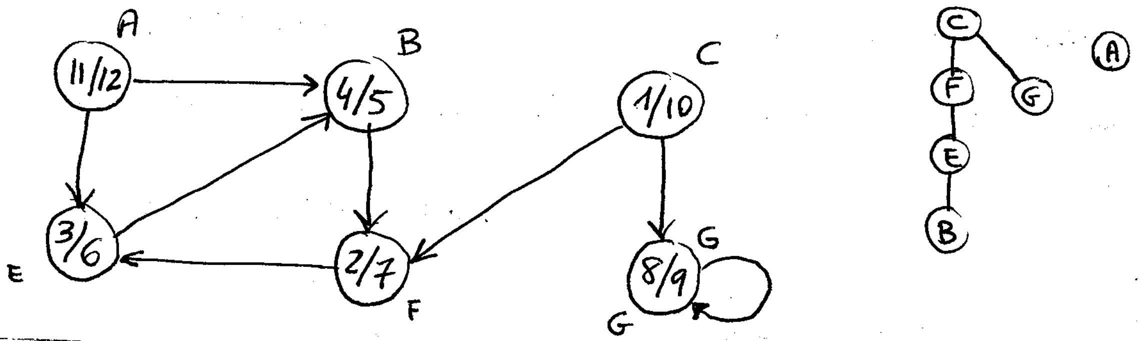
$Q = \{H_3\}$

$Q = \{\}$

Búsqueda en profundidad

NODO [td / tf] → finishing time





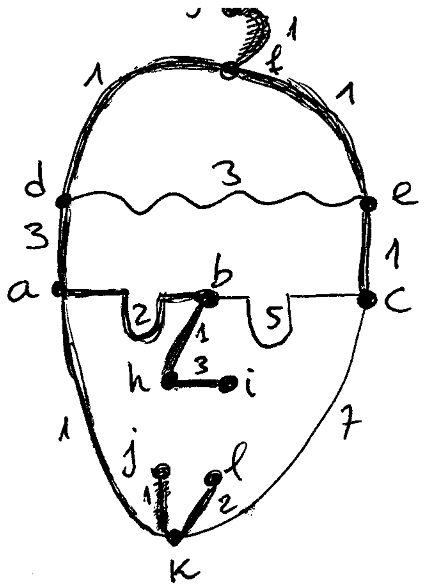
ÁRBOLES ABARCADORES de COSTE MÍNIMO

KRUSKAL → Siempre mínimo (no necesariamente conexo) → en pasos intermedi

1. Ordena aristas de menor a mayor peso.
2. Incorpora las aristas una a una, asegurándonos que no hay ciclos
3. Sumar los pesos (peso mínimo resultante).

PRIM → tiene que ser conexo.

0. Partiendo de un nodo
1. Incluye enlaces de peso mínimo que mantengan subgrafo <u>conexo</u> y <u>no</u> formen <u>ciclos</u>.



12 nodos

ÁRBOL ABARCADOR

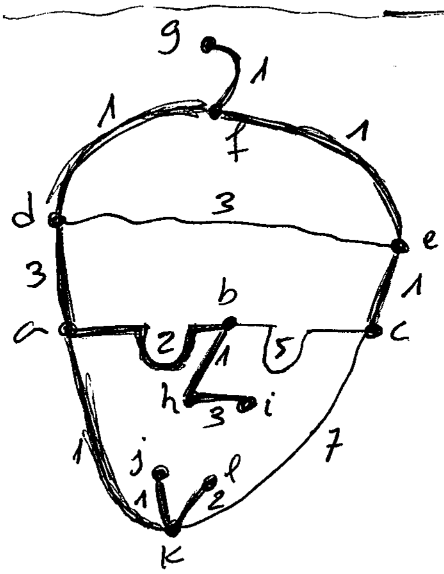
¿cuantos enlaces?

11 enlaces

PESO MÍNIMO $\boxed{\text{peso } 17}$

ARISTA	PESO	¿INCLUIR EN ÁRBOL?
$\overline{AK}$	1	Sí
$\overline{BH}$	1	Sí
$\overline{CE}$	1	Sí
$\overline{DF}$	1	Sí
$\overline{EF}$	1	Sí
$\overline{FG}$	1	Sí
$\overline{JK}$	1	Sí
$\overline{AB}$	2	Sí
$\overline{KL}$	2	Sí
$\overline{AD}$	3	Sí
$\overline{DE}$	3	No
$\overline{HI}$	3	Sí
$\overline{BC}$	5	No
$\overline{CK}$	7	No

ALGORITMO DE KRUSKAL



12 nodos

ÁRBOL ABARCADOR

¿cuantos enlaces?

11 enlaces

PESO MÍNIMO $\boxed{\text{peso } 17}$

ARISTA	PESO	¿INCLUIR EN ÁRBOL?
$\overline{AK}$	1	1 Sí
$\overline{BH}$	1	4 Sí
$\overline{CE}$	1	9 Sí
$\overline{DF}$	1	7 Sí
$\overline{EF}$	1	8 Sí
$\overline{FG}$	1	10 Sí
$\overline{JK}$	1	2 Sí
$\overline{AB}$	2	3 Sí
$\overline{KL}$	2	5 Sí
$\overline{AD}$	3	6 Sí
$\overline{DE}$	3	
$\overline{HI}$	3	11 Sí
$\overline{BC}$	5	
$\overline{CK}$	7	

ALGORITMO DE PRIM

ALGORITMO (parte) DE BÚSQUEDA EN ANCHURA

distancia; predecesor

$d[v] \leftarrow 0$; $color[v] \leftarrow WHITE$; $d[v] \leftarrow \infty$; $\pi[v] \leftarrow NIL$ $\forall v \in V$.

$Q \leftarrow \phi$; ENQUEUE (Q, s) ; $d[s] \leftarrow 0$; $color[s] \leftarrow GRAY$

while $Q \neq \phi$

    do $u \leftarrow$ dequeue (Q)

        for each $v \in Adj.[u]$

            do if $color[v] = WHITE$

                then $color[v] \leftarrow GRAY$

                    $d[v] \leftarrow d[u] + 1$

                    $\pi[v] \leftarrow u$

                    ENQUEUE (v)

        $color[u] \leftarrow BLACK$

S: nodo inicial (correspondiente a la raíz del árbol BFS)

$d[v]$: etiqueta numérica del vértice

$\delta(s,v) \rightarrow$ longitud de la trayectoria más corta entre S y v.
$= \infty$ si S y v no están conectados

$G \equiv \{V, E\}$

#vértices $|V|$; #aristas $|E|$

de ops. ENQUEUE $= |V|$

de ops. DEQUEUE $= |V|$

ops. de búsqueda en lista de adyacencia $= 2|E|$

$$2|V| + 2|E| = \Theta(|V| + |E|) = \text{COMPLEJIDAD}$$

INVARIANTE DE BUCLE:

Los nodos en Q están en GRAY.

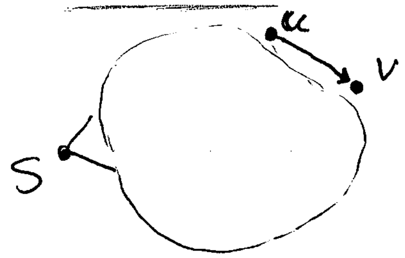
LEMA (En el CORMEN Lema 22.1)

$\forall (u,v) \in E$ $\boxed{\delta(s,v) \le \delta(s,u) + 1}$

del que parte la flecha

al que apunta la flecha

Demostración



1^er caso

Si s y u no están conectados $\left(\delta(s,u) = \infty\right)$

$\Downarrow$

s y v no están conectados $\left(\delta(s,v) = \infty\right)$

ó

$\underset{\infty}{\delta(s,v)} = \underset{\infty + 1}{\delta(s,u) + 1}$ ✓

están conectados

$\delta(s,v) < \infty \Rightarrow$

$\Rightarrow \underbrace{\delta(s,v)}_{<\infty} < \underbrace{\delta(s,u)}_{=\infty} \le \underbrace{\delta(s,u) + 1}_{=\infty + 1} \Rightarrow$

$\Rightarrow \delta(s,v) < \delta(s,u) + 1$ ✓

$\Rightarrow \delta(s,v) \le \delta(s,u) + 1$

2º caso

Si s y u están conectados $\Rightarrow$ s y v están conectados $\Rightarrow$

$\Rightarrow \delta(s,v) < \infty$

- Supongamos que la trayectoria más corta de s a v sea pasando por u $s \rightsquigarrow u \longrightarrow v$ $\Rightarrow \delta(s,v) = \delta(s,u) + 1$
- Supongamos que la trayectoria más corta de s a v no incluya a u $\Rightarrow \delta(s,v) < \delta(s,u) + 1$

$\Rightarrow \delta(s,v) \le \delta(s,u) + 1$

LEMA (En el CORMEN 22.2)

Cuando BFS termina $\boxed{d[v] \geq \delta(s,v)}$; $\forall v \in V$

$d[v]$ → etiqueta; $\delta(s,v)$ → camino óptimo

Demostración : POR INDUCCIÓN en ENQUEUE

- CASO BASE:

Al principio del algoritmo $d[v] = \infty$ $\forall v \in V$.

$\Downarrow$

$d[v] \geq \delta(s,v)$

$d[s] = 0$ $\qquad \delta(s,s) = 0$

$d[s] = \delta(s,s)$ se cumple ; y también : $d[s] \geq \delta(s,s)$

- Supongamos que → vamos a explorar $u \leftarrow$ DEQUEUE (Q)
  ↳ antes de explorar u $^{(*)}$ $d[v] \geq \delta(s,v)$ $\forall v \in V$.

Consideremos $v \in Adj[u]$: (tiene que haber enlace $u \rightarrow v$).

- Si v no es WHITE : $d[v]$ no cambia $\Rightarrow d[v] \geq \delta(s,v)$
- Si v es WHITE : $d[v] = d[u] + 1 \overset{(*)}{\geq} \delta(s,u) + 1 \overset{(**)}{\geq} \delta(s,v)$ $\Longrightarrow$

(**) Por el LEMA 22.1 $\delta(s,v) \leq \delta(s,u) + 1$

$\Longrightarrow d[v] \geq \delta(s,v)$

LEMA (En el CORMEN Lema 22.3)

Supongamos que en un punto intermedio de BFS: $Q = \{V_1, V_2, \dots, V_r\}$

(i) $d[V_r] \leq d[V_1] + 1$

(ii) $d[V_i] \leq d[V_{i+1}]$; $i = 1, 2, \dots, r-1$

Demostración

- CASO BASE: $Q \leftarrow \phi$

1ª interacción: $Q \leftarrow \{s\}$

$d[\phi] = 0$
$d[s] = 0$

$d[\phi] = d[s]$
$\Downarrow$
$d[\phi] \leq d[s]$ ✓

- CASO GENERAL:

Supongamos que $Q = \{V_1, V_2, \dots, V_r\}$ cumple (i) y (ii):

DEQUEUE(Q) $\Rightarrow Q = \{V_2, V_3, \dots, V_r\}$; $u \leftarrow V_1$

$$\left.\begin{array}{l}\text{(i) } d[V_r] \leq d[V_1] + 1 \\ \text{(ii) } d[V_1] \leq d[V_2]\end{array}\right\} \Rightarrow d[V_r] \leq d[V_1] + 1 \leq d[V_2] + 1 \Rightarrow$$

$$\Rightarrow \boxed{d[V_r] \leq d[V_2] + 1} \ (*) \quad ✓$$

(ii) $d[V_1] \leq d[V_2] \leq d[V_3] \leq \dots \leq d[V_{r-1}] \leq d[V_r]$

$\Rightarrow d[V_2] \leq d[V_3] \leq \dots \leq d[V_r]$

$d[V_2] \leq d[V_3] \leq \dots \leq d[V_r] \equiv \boxed{d[V_i] \leq d[V_{i+1}] \ ; \ i = 2, 3, \dots, r}$ (**) ✓

¿Pero, qué ocurre tras ENQUEUE? $\underbrace{V_{nuevo} \in Adj[V_1]}$ $\Rightarrow \boxed{d[V_{nuevo}] = d[V_1] +}$ (***

- Además, debe haber enlace $V_1 \rightarrow V_{nuevo}$

ENQUEUE(Q, V_{nuevo}) $\quad Q = \{V_2, V_3, \dots, V_r, V_{nuevo}\}$

$d[V_{nuevo}] = d[V_1] + 1 \overset{(i)}{\geq} d[V_r] \Rightarrow \boxed{d[V_r] \leq d[V_{nuevo}]}$ (***)

(**) + (***) $\Rightarrow \boxed{d[V_2] \leq d[V_3] \leq \dots \leq d[V_r] \leq d[V_{nuevo}]}$ ✓ por (ii)

$d[V_{nuevo}] \leq d[V_2] + 1$?

$d[V_{nuevo}] = d[V_1] + 1$; $d[V_2] \geq d[V_1]$; $d[V_2] + 1 \geq d[V_1] + 1$;

$$\boxed{d[V_{nuevo}] \leq d[V_2] + 1}$$

COROLARIO 22.4

Si V_i es introducido en la cola antes que V_j:

$$d[V_i] \leq d[V_j]$$

TEOREMA 22.5

BFS descubre todos los vértices accesibles desde S

$\forall v \in V$; $d[v] = \delta(s,v)$ al final del algoritmo BFS

Demostración (reducción al absurdo/contradicción)

NEGACIÓN DE LA META

Supongamos que $\exists v \in V$, accesible desde S, para el cual $d[v] > \delta(s,v)$ al final del algoritmo BFS (*)

→ teniendo en cuenta L22.2

① Si v no es accesible desde S: $\delta(s,v) = \infty$ (por def. de "ser accesible").

Por otra parte, si v no es accesible desde s ⇒

⇒ Nunca se actualiza $d[v]$ (que valdrá ∞)

$d[v] = \delta(s,v)$ ⇒ Luego tiene que ser accesible

② Como v es accesible desde S: $\delta(s,v) < \infty$

Consideremos el enlace $u \to v$, donde u es el predecesor de v en el camino más corto entre $s \rightsquigarrow v$.

Para este u sí se cumple $\boxed{d[u] = \delta(s,u)}$ (**)

$$\left.\begin{array}{l} \delta(s,v) = \delta(s,u) + 1 \\ (*)\ d[v] > \delta(s,v) \end{array}\right\} \Longrightarrow d[v] > \delta(s,u) + 1 \overset{(**)}{\Longrightarrow} \boxed{d[v] > d[u] + 1}\ (***)$$

¿qué ocurre cuando BFS saca a u de la cola Q?

CASO 1: V es BLANCO: $d[v] = d[u] + 1$; en contradicción con (***) → DESCARTADO

CASO 2: V es NEGRO: v estaba en la cola antes que u $\overset{\text{COLORARIO 22.4}}{\Longrightarrow}$ $d[v] \leq d[u]$; en contradicción con (***) → DESCARTADO

CASO 3: V es GRIS: $Q = \{\underset{\text{sacando}}{u}, \ldots, v, \ldots\}$ de acuerdo con L22.3 (i): $d[V_{último}] \leq d[u] + 1$

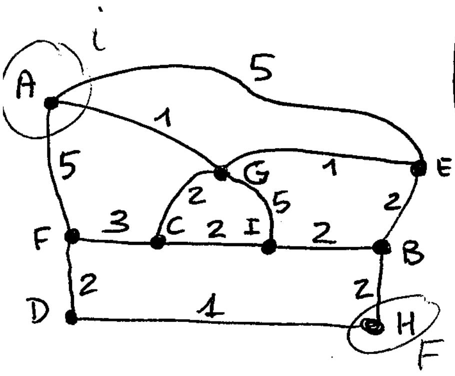
de acuerdo con L22.3 (ii) $d[v] \leq d[V_{último}]$

⇒ $d[v] \leq d[u] + 1$; en contradicción con (***) → DESCARTADO

Entonces la hipótesis (*) es ERRÓNEA ⇒ $\boxed{\text{BASE CONOCIMIENTO + META NEGADA es contradictoria}}$,

por lo que queda demostrado: $d[v] = \delta(s,v)$ al final de BFS

EJERCICIO 1



	L_0	L_1	L_2	L_3	L_4	L_5	L_6	L_7
A	(0)*	–	–	–	–	–	–	–
B	∞	∞	∞	4e	(4e)*	–	–	–
C	∞	∞	3g	(3g)*	–	–	–	–
D	∞	∞	∞	∞	∞	∞	7f	7f
E	∞	5a	(2g)*	–	–	–	–	–
F	∞	5a	5a	5a	5a	(5a)*	–	–
G	∞	(1a)*	–	–	–	–	–	–
H	∞	∞	∞	∞	∞	6b	6b	(6b)*
I	∞	∞	6g	6g	5c	5c	(5c)*	–

~~DISTANCIA MÍNIMA: 6~~

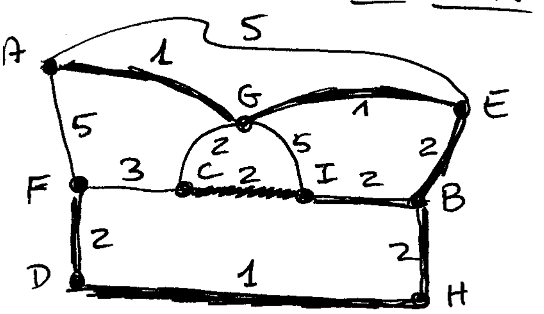
PESO MÍNIMO
6

h←b←e←g←a

CAMINO ÓPTIMO

EJERCICIO 2

ALGORITMO DE PRIM

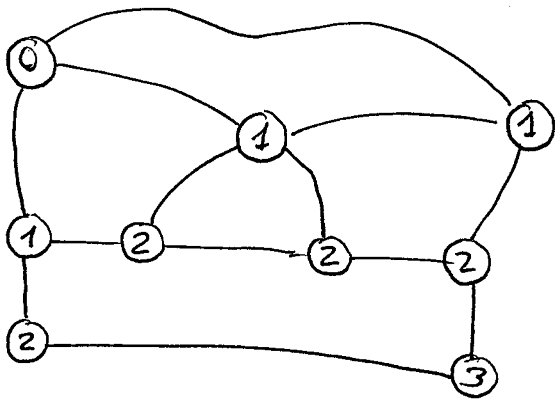


No necesitamos examinar más aristas porque ya se han alcanzado todos los nodos.

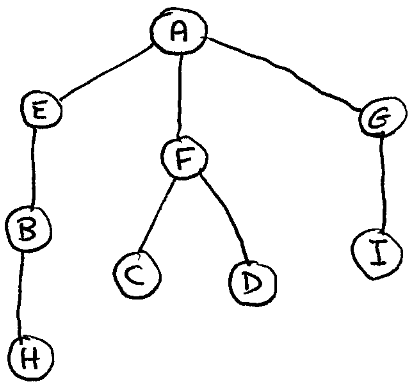
ARISTA EXAMINADA	PESO	¿E/D?
A - G	1	E
G - E	1	E
E - B	2	E
B - H	2	E
H - D	1	E
B - I	2	E
D - F	2	E
I - C	2	E

PESO TOTAL MÍNIMO: 13

EJERCICIO 3



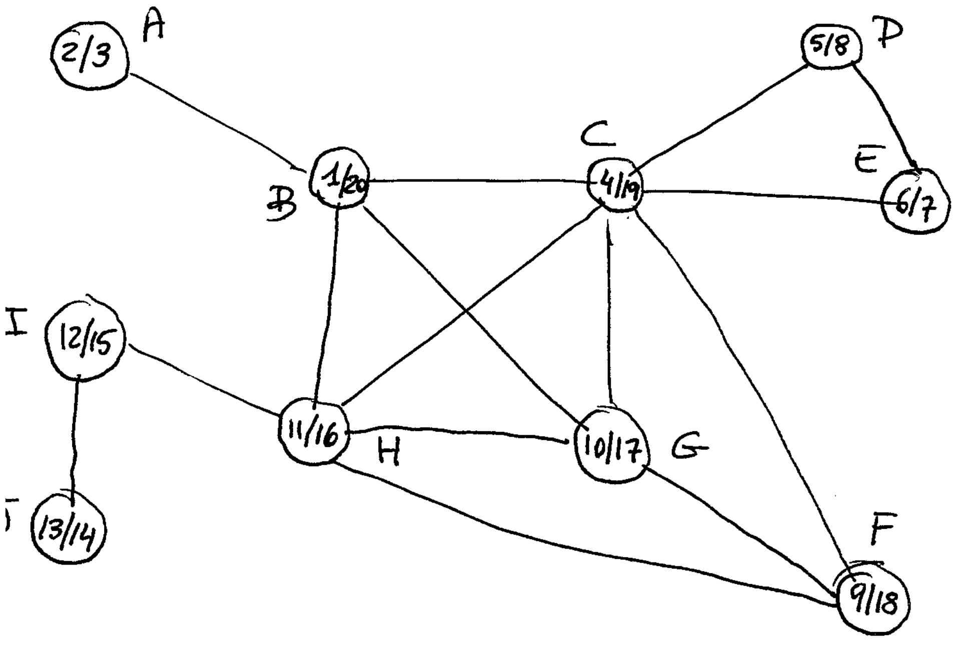
ÁRBOL RESULTANTE:



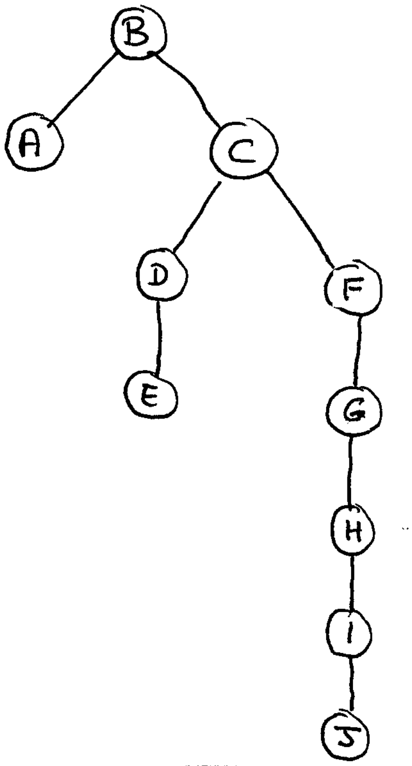
ORDE DE EXPLORACIÓN:

A - E - F - G - B - C - D - I

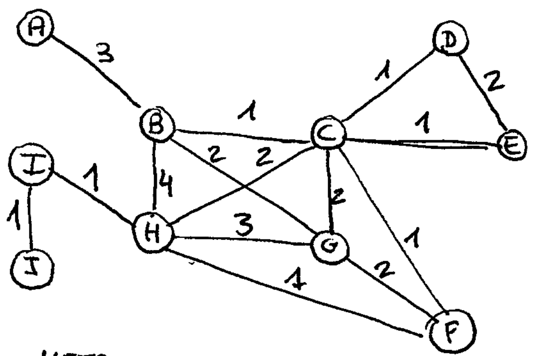
EJERCICIO 4



ÁRBOL RESULTANTE:



EJERCICIO 5



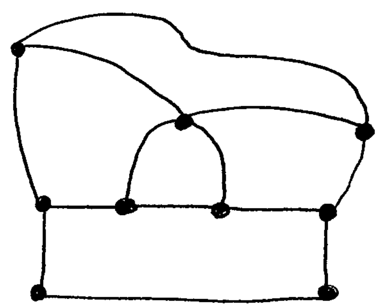
META — INICIO

H ← C ← B

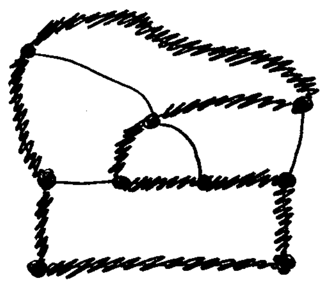
	L_0	L_1	L_2	L_3	L_4	L_5	L_6
A	∞	3b	3b	3b	3b	(3b)*	—
B	(0)*	—	—	—	—	—	—
C	∞	(1b)*	—	—	—	—	—
D	∞	∞	(2c)*	—	—	—	—
E	∞	∞	2c	(2c)*	—	—	—
F	∞	∞	∞	∞	∞	4g	4g
G	∞	2b	2b	2b	(2b)*	—	—
H	∞	4b	3c	3c	3c	3c	(3c)*
I	∞	∞	∞	∞	∞	∞	∞
J	∞	∞	∞	∞	∞	∞	∞

DISTANCIA MÍNIMA: 3

EJERCICIO 8

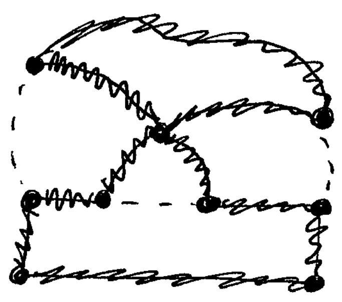


Sí que contiene un circuito hamiltoniano:



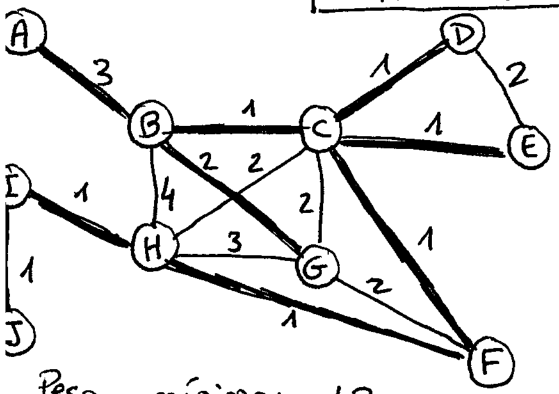
No contiene circuitos eulerianos ya que no todos sus vértices tienen grado par.

Para que sí que tenga, eliminamos 3 aristas:



EJERCICIO 6

Algoritmo de KRUSKAL

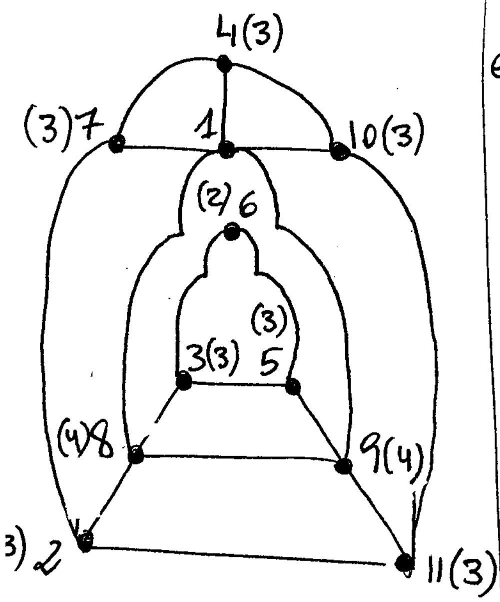


Peso mínimo: 12

- No necesitamos examinar más aristas porque todos los nodos ya han sido alcanzados.

ARISTA EXAMINADA	PESO	ELEGIDA/DESCARTADA
B-C	1	E
C-D	1	E
C-E	1	E
C-F	1	E
F-H	1	E
H-I	1	E
I-J	1	E
B-G	2	E
C-G	2	D
C-H	2	D
D-E	2	D
G-F	2	D
A-B	3	E
G-H	3	
B-H	4	

EJERCICIO 7

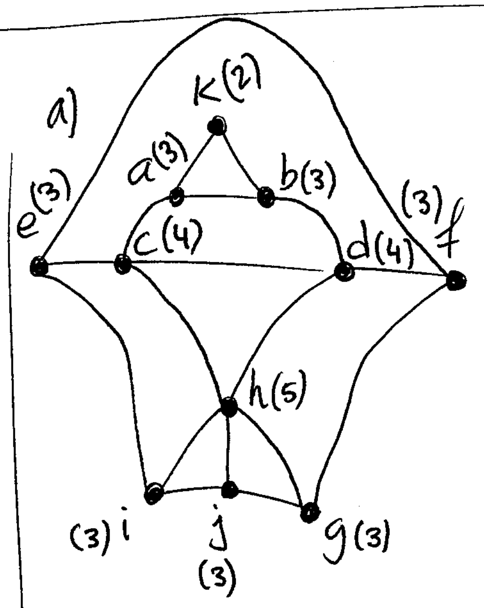


nº nodos: 11

nº enlaces: 18

grados
- 7 vértices (3)
- 1 vértice (2)
- 2 vértices (4)
- 1 vértice (5)

a)



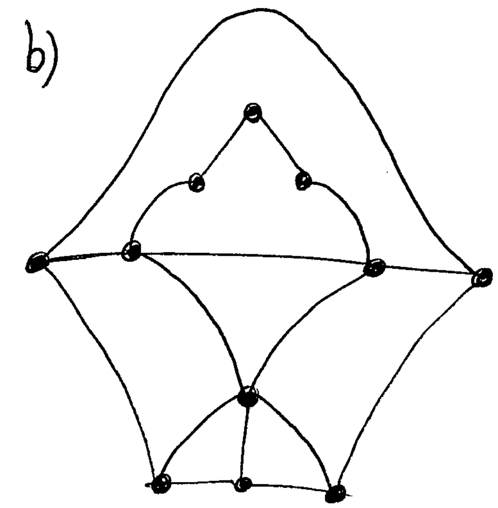
nº nodos: 11

nº enlaces: 18

grados
- 7 vértices (3)
- 1 vértice (2)
- 2 vértices (4)
- 1 vértice (5)

$f(1) = h$	$f(6) = k$
$f(2) = e$	$f(7) = i$
$f(3) = a$	$f(8) = c$
$f(4) = j$	$f(9) = d$
$f(5) = b$	$f(10) = g$
	$f(11) = f$

b)



nº nodos: 11

nº enlaces: 17

No es isomorfo porque no respeta uno de los invariantes que existen bajo un isomorfismo (el nº de enlaces no es igual)