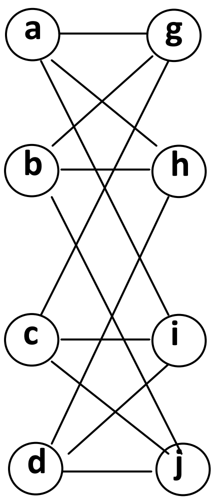
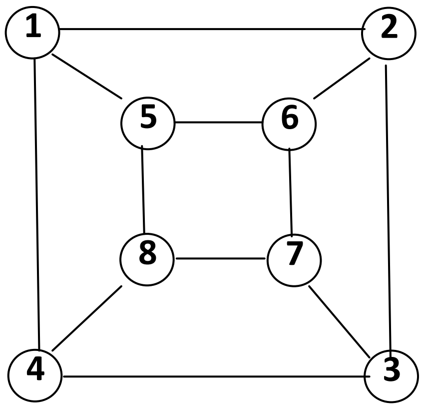
HOJA DE EJERCICIOS 4: Grafos y Árboles EDyL 2011-2012

[Fecha de publicación: 21 noviembre 2011]
[Fecha de entrega: 5 diciembre 2011, 10 horas]
[Resolución en clase 5 diciembre 2011]

EJERCICIO 1: Determinar si los pares de grafos siguientes son o no isomorfos. Justificar las respuestas.

a) Grafo G Grafo H



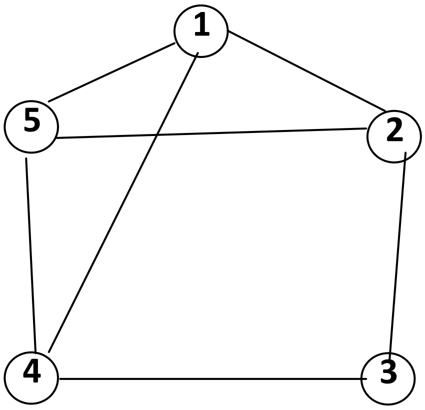


RESPUESTA: G y H son isomorfos. Una posible función que relaciona los vértices de uno y otro grafo es la siguiente:

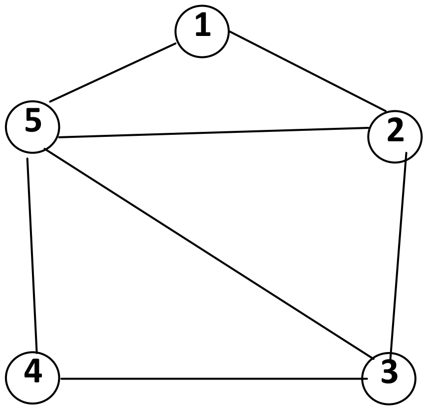
f(a) = 1
f(b) = 6
f(c) = 8
f(d) = 3
f(g) = 5
f(h) = 2
f(i) = 4
f(j) = 7

También se puede verificar mediante las correspondientes matrices de adyacencia.

b) Grafo G



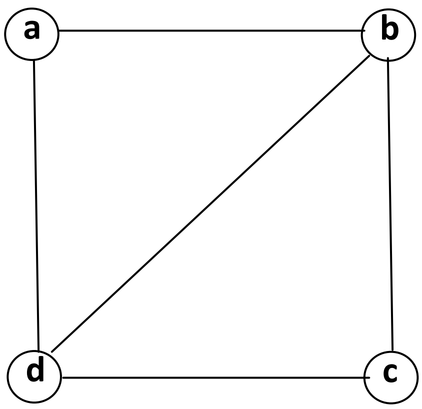
Grafo H



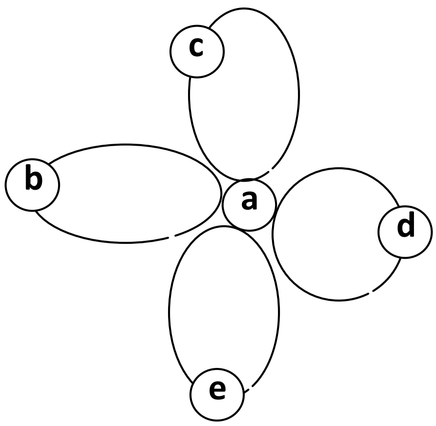
Estos dos grafos no son isomórficos ya que, aunque tienen el mismo número de vértices y de aristas, el grado de los vértices adyacentes en cada uno de los grafos no coincide.

EJERCICIO 2: Determinar la existencia de circuitos o caminos eulerianos en los grafos siguientes. Justificar las respuestas.

G1



G2

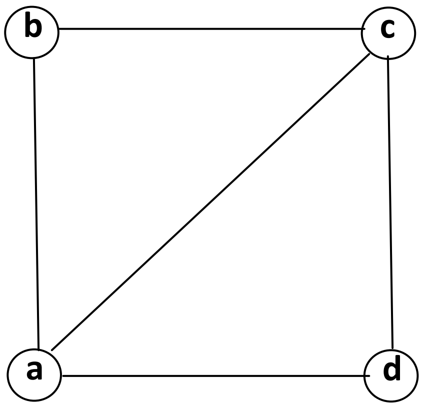


G1 contiene una trayectoria euleriana entre b y d: b-d-c-b-a-d

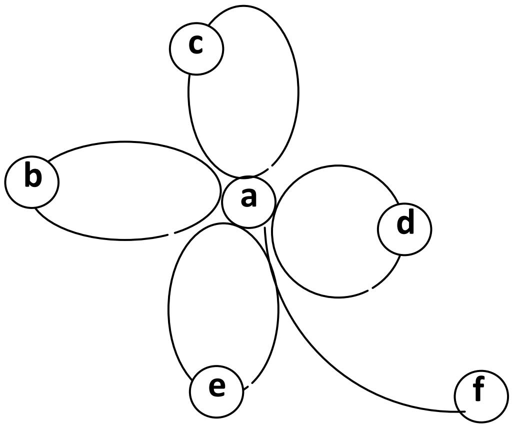
G2 contiene un circuito euleriano, por lo que es un grafo euleriano: a-b-a-c-a-d-a-e-a

EJERCICIO 3: Determinar la existencia de circuitos o caminos hamiltonianos en los grafos siguientes. Justificar las respuestas.

G1



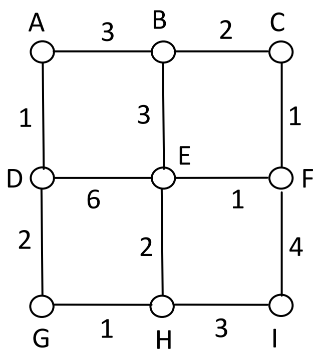
G2



G1 contiene un circuito hamiltoniano: a-b-c-d-a

G2 no contiene ni trayectorias ni circuitos hamiltonianos.

EJERCICIO 4: Dado el siguiente grafo, emplear el algoritmo de Dijkstra para encontrar la distancia más corta entre los nodos A y E, indicando a qué trayectoria corresponde. Utilizar tantas columnas de la tabla como sea necesario.

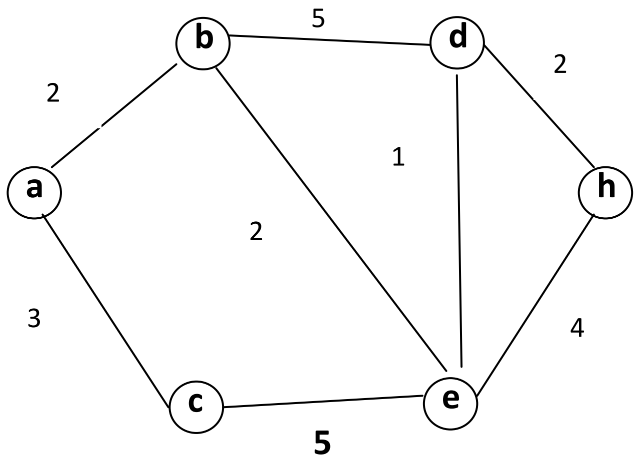


SOLUCIÓN:

	L(0)	L(1)	L(2)	L(3)	L(4)	L(5)	L(6)
A	(0)	-		-	-	-	-
B	∞	B_A 3	B^*_A (3)	-	-	-	-
C	∞	∞	∞	C_B 5	C_B 5	C^*_B (5)	-
D	∞	D^*_A (1)	-	-	-	-	-
E	∞	∞	E_D 7	E_B 6	E_B 6	E_B 6	E^*_B (6)
F	∞	∞	∞	∞	∞	∞	F_C 6
G	∞	∞	G_D 3	G^*_D (3)	-	-	-
H	∞	∞	∞	∞	H^*_G (4)	-	-
I	∞	∞	∞	∞	∞	I_H 7	I_H 7

La distancia más corta entre A y E es 6, y la trayectoria: A-B-E.

EJERCICIO 5: Utilizando el algoritmo de Warshall, determinar las distancias más cortas entre cada par de vértices del grafo, así como las trayectorias correspondientes.



L0	a	b	c	d	e	h
a	0	2	3	∞	∞	∞
b		0	∞	5	2	∞
c			0	∞	5	∞
d				0	1	2
e					0	4
h						0

L1 (a)	a	b	c	d	e	h
a	0	2(ab)	3(ac)	∞	∞	∞
b		0	5(bac)	5(bd)	2(be)	∞
c			0	∞	5(ce)	∞
d				0	1(de)	2(dh)
e					0	4(eh)
h						0

L2 (b)	A	b	c	d	e	h
a	0	2(ab)	3(ac)	7(abd)	4(abe)	∞
b		0	5(bac)	5(bd)	2(be)	∞
c			0	10(cabd)	5(ce)	∞
d				0	1(de)	2(dh)
e					0	4(eh)
h						0

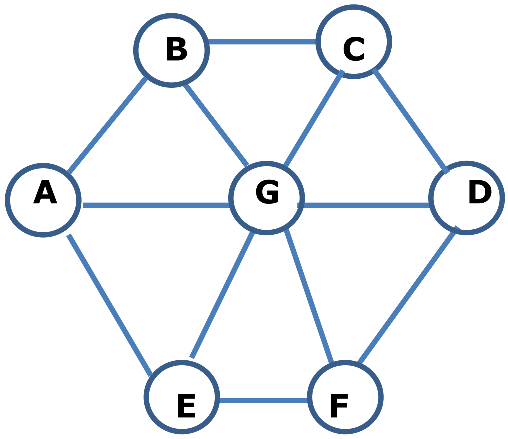
L3 (c)	A	b	c	d	e	h
a	0	2(ab)	3(ac)	7(abd)	4(abe)	∞
b		0	5(bac)	5(bd)	2(be)	7(bdh)
c			0	10(cabd)	5(ce)	∞
d				0	1(de)	2(dh)
e					0	4(eh)
h						0

L4 (d)	**A**	**b**	**c**	**d**	**E**	**h**
a	0	2(ab)	3(ac)	7(abd)	4(abe)	9(adh)
b		0	5(bac)	5(bd)	2(be)	7(bdh)
c			0	10(cabd)	5(ce)	12(cabdh)
d				0	1(de)	2(dh)
e					0	3(edh)
h						0

L5 (e)	**A**	**b**	**c**	**d**	**E**	**H**
a	0	2(ab)	3(ac)	5(abed)	4(abe)	7(abedh)
b		0	5(bac)	3(bed)	2(be)	5(bedh)
c			0	6(ced)	5(ce)	8(cedh)
d				0	1(de)	2(dh)
e					0	3(edh)
h						0

L6 (h)	**A**	**b**	**c**	**d**	**E**	**H**
a	0	2(ab)	3(ac)	5(abed)	4(abe)	7(abedh)
b		0	5(bac)	3(bed)	2(be)	5(bedh)
c			0	6(ced)	5(ce)	8(cedh)
d				0	1(de)	2(dh)
e					0	3(edh)
h						0

EJERCICIO 6: Dado el siguiente grafo, emplear el algoritmo de Búsqueda en Profundidad para descubrir todos los nodos accesibles desde A. Incluir las etiquetas de tiempo inicial y final de exploración para cada nodo accedido.



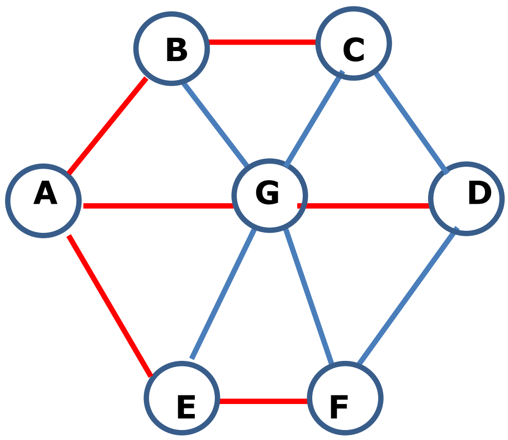
SOLUCiÓN:

Búsqueda en profundidad Cormen

A (1/) B(2 /) C(3/) D(4/) F(5/) E(6/) G(7/) G(7/8) E(6/9) F(5/10)
D(4/11) C(3/12) B(2/13) A (1/14)

EJERCICIO 7: Para el grafo del ejercicio anterior, emplear el algoritmo de Búsqueda en Anchura para descubrir todos los nodos accesibles desde A.

SOLUCiÓN:



Búsqueda en anchura Cormen

A

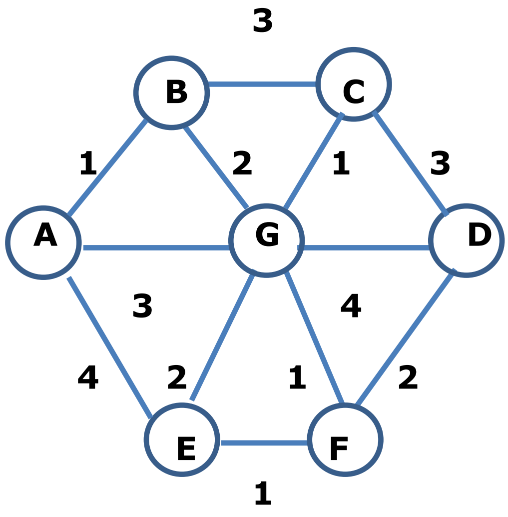
B E G

C

F

D

EJERCICIO 8: Hallar un árbol extendido mínimo para el grafo siguiente, utilizando el algoritmo de Kruskal.



SOLUCIÓN:

Aristas por orden creciente de peso (y orden alfabético):

arista 1	A-B: 1
arista 2	C-G: 1
arista 3	E-F: 1
arista 4	F-G: 1
arista 5	B-G: 2
arista 6	D-F: 2
arista 7	E-G: 2
arista 8	A-G: 3
arista 9	B-C: 3
arista 10	C-D: 3
arista 11	A-E: 4
arista 12	F-G: 4

El orden en el que incluyen aristas y nodos es el siguiente:

1) Incluir A-B (arista 1), se acceden A y B

2) Incluir C-G (arista 2), se acceden C y G

3) Incluir E-F (arista 3), se acceden E y F

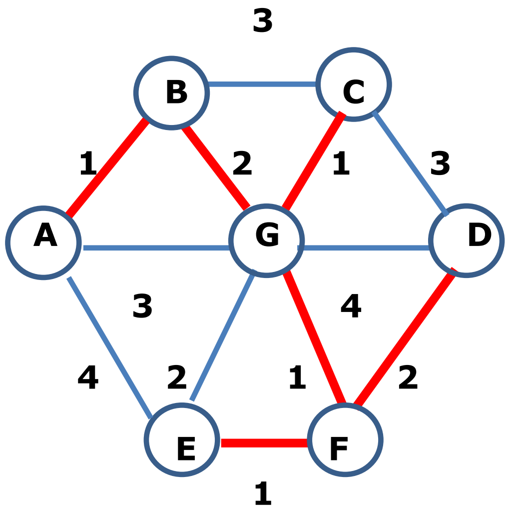
4) Incluir F-G (arista 4), se accede G (F ya se había accedido)

5) Incluir B-G (arista 5), se conectan B y G, aunque no se accede ningún nuevo nodo

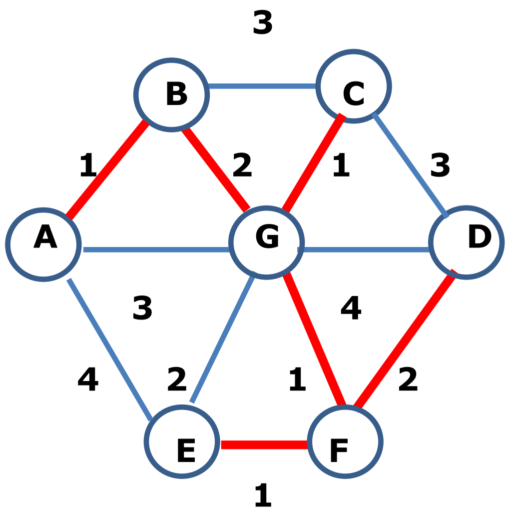
6) Incluir D-F (arista 6), se accede D (F ya se había accedido)

El algoritmo concluye habiéndose accedido ya a todos los nodos, y el resto de las aristas NO se consideran.

El árbol resultante se representa a continuación, y su peso es 8.



EJERCICIO 9: Hallar un árbol extendido mínimo para el grafo del ejercicio anterior utilizando el algoritmo de Prim.



Selección:

AB (1) Selecciona Vértices={A, B}

Posibilidades (adyacentes a A o B):

BG	(2)	Selecciona (adyacente a B)	Vértices={A, B, G}
AG	(3)		
BC	(3)		
AE	(4)		

Posibilidades (adyacentes a A, B o G):

CG	(1)	Selecciona (adyacente a G)	Vértices={A, B, C, G}
FG	(1)		
EG	(2)		
AG	(3)		
BC	(3)		
AE	(4)		
DG	(4)		

Posibilidades (adyacentes a A, B, C o G):

FG (1) Selecciona (adyacente a G) Vértices={A, B, C, F, G}

EG (2)

AG (3)

BC (3)

CD (4)

AE (4)

DG (4)

Posibilidades (adyacentes a A, B, C, F o G):

EF (1) Selecciona (adyacente a F) Vértices={A, B, C, E, F, G}

DF (2)

EG (2)

AG (3)

BC (3)

CD (4)

AE (4)

DG (4)

Posibilidades (adyacentes a A, B, C, E, F o G):

DF (2) Selecciona (adyacente a F) Vértices={A, B, C, D, E, F, G}

EG (2)

AG (3)

BC (3)

CD (4)

AE (4)

DG (4)

Peso = 8

EJERCICIO 10: Demostrar, por inducción, que un árbol con n vértices posee n-1 aristas.

SOLUCIÓN:

Base: es cierto para n=1, 2, 3…

Hipótesis: es cierto para todos los árboles con menos de n vértices (n-1 vértices es el último para el que es cierto)

Inducción: se elimina una arista de T $\Rightarrow$ quedan dos componentes conexas sin circuitos: T1, con r vértices, y T2, con n-r vértices.
Por la hipótesis, T1 tiene (r-1) aristas y T2 tiene (n-r-1).
Luego T tiene (r-1)+(n-r-1) +1=n-1