HOJA DE EJERCICIOS 4: Grafos y Árboles **EDyL 2013-2014**

[Fecha de publicación: 2013/11/20]
[Fecha de entrega: 2013/12/03, 09:00]
[Resolución en clase: 2013/12/03, 2013/12/05]

EJERCICIO 1 [adaptado de Veerarajan, Cap. 7, Ej. 37]: Dibuja, en caso de que sea posible, grafos simples no dirigidos cuyas listas de grados de vértices sean las que se indican a continuación. En caso de que no sea posible el grafo, indica la razón para ello.

(a) {1,2,3,4,5}

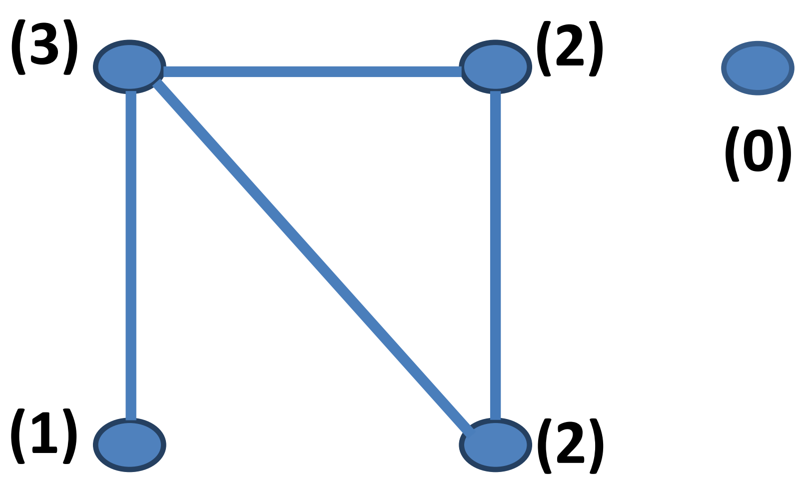
No es posible: la suma de los grados de los vértices es impar.

(b) {1,2,3,4,4}

No es posible: Dado que no hay bucles, los dos nodos de grado 4 tienen que estar conectados a cada uno de los otros cuatro nodos del grafo. Por lo tanto los nodos del grafo deberían ser de grado 2 o superior.

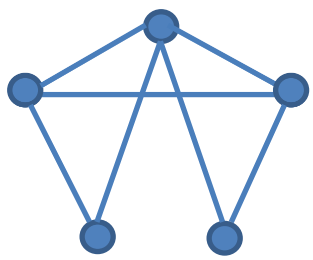
(c) {3,4,3,4,3}
No es posible: la suma de los grados de los vértices es impar.

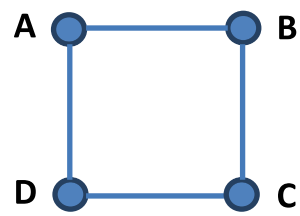
(d) {0,1,2,2,3}



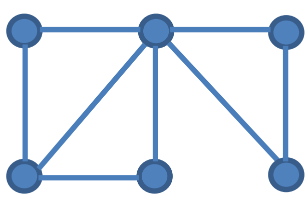
EJERCICIO 2 [adaptado de Veerarajan, Cap. 7, Ej. 41]: Para cada una de los casos, indica si el grafo de la derecha es subgrafo del de la izquierda. En caso de que no sea subgrafo indica por qué razón. Si es subgrafo, marca los vértices correspondientes en el grafo original con las etiquetas de los vértices del subgrafo

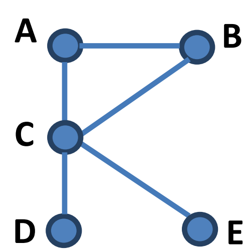
(a)





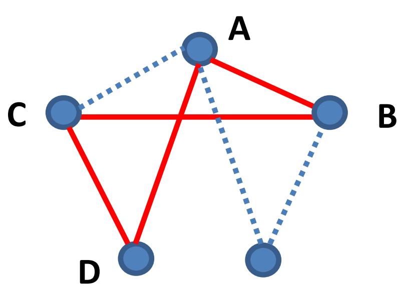
(b)

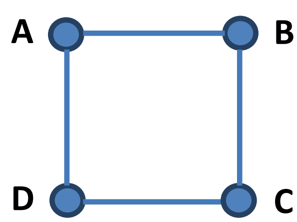




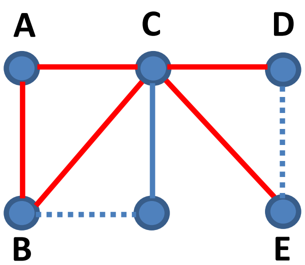
SOLUCIÓN:

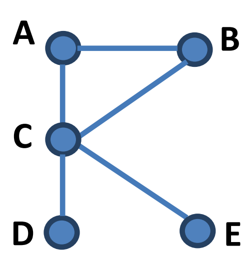
(a) Es subgrafo.





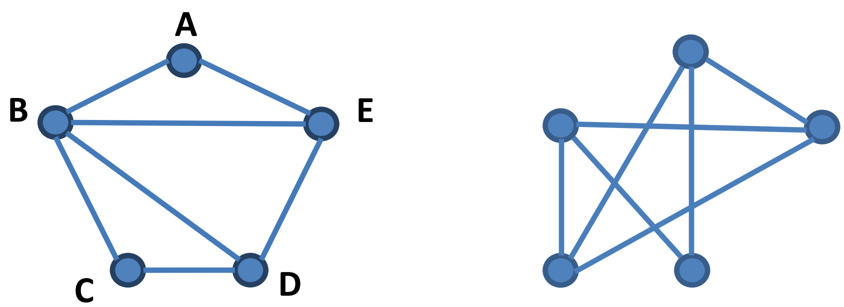
(b) Es subgrafo



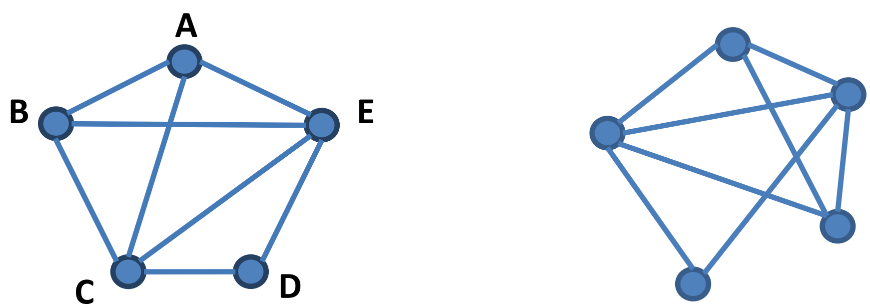


EJERCICIO 3 [adaptado de Veerarajan, Cap. 7, Ej. 42]: Indica si son isomorfos los siguientes grafos. En caso de que no lo sean, explica qué razones te han llevado a dicha conclusión. En caso que lo sean, etiqueta los vértices del grafo de la derecha de forma que se ponga de manifiesto el isomorfismo.

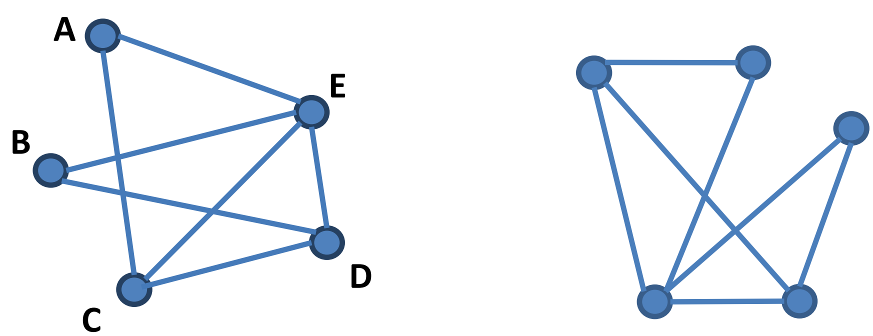
(a)



(b)

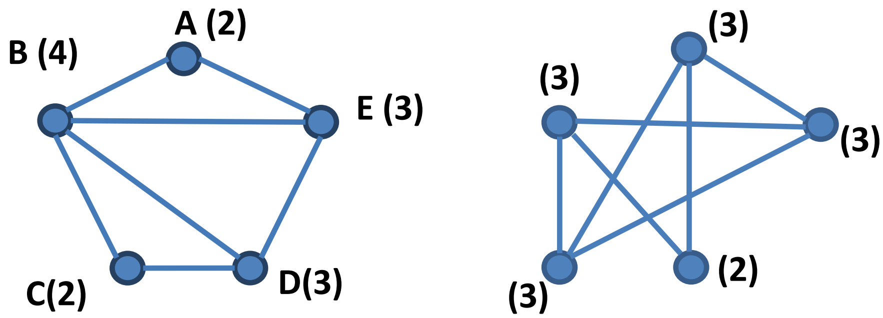


(c)



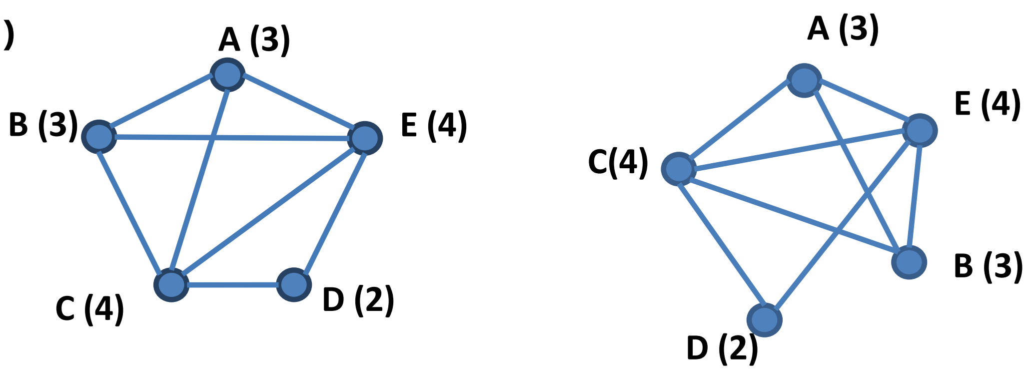
SOLUCIÓN:

(a)



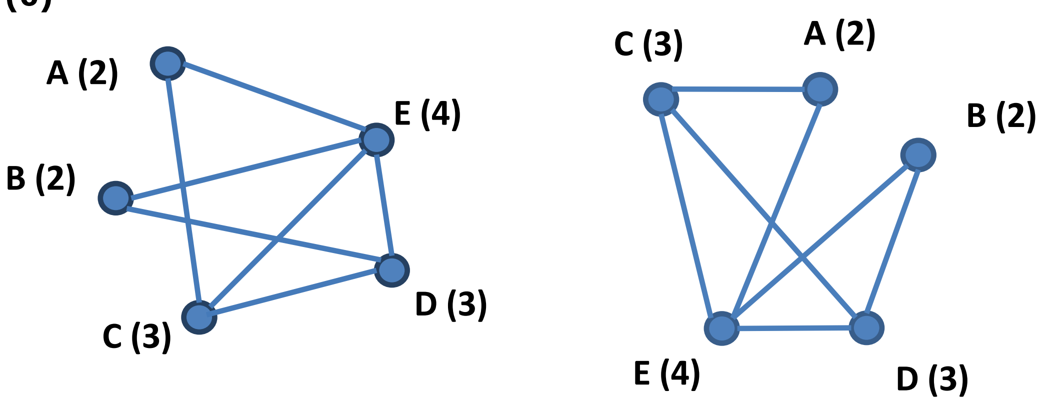
No son isomorfos ya que los grados de los nodos son distintos.

(b)



Isomorfos

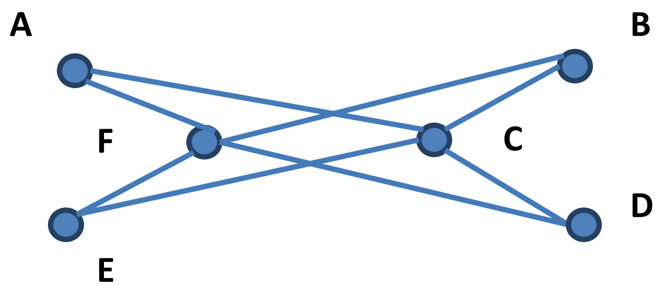
(c)



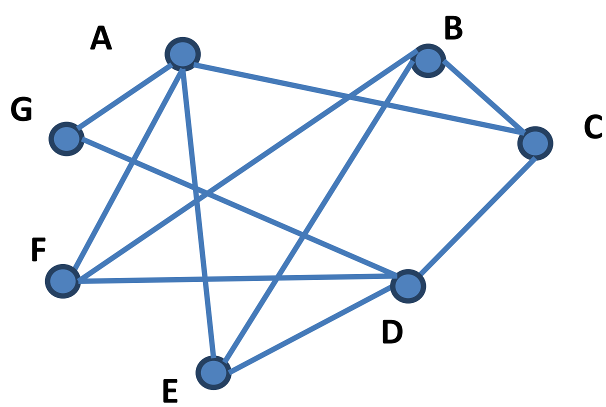
Isomorfos

EJERCICIO 4 [adaptado de Veerarajan, Cap. 7, Ej. 39]: Indica si alguno de los siguientes grafos son bipartitos. En caso de que lo sean, indica si son bipartitos completos.

(a)

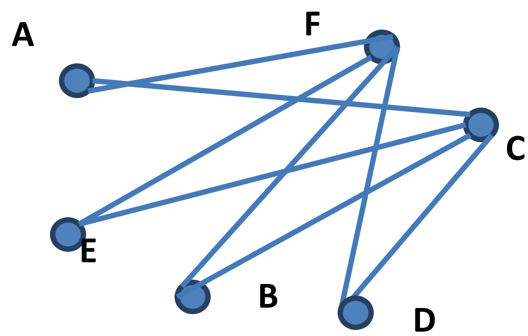


(b)

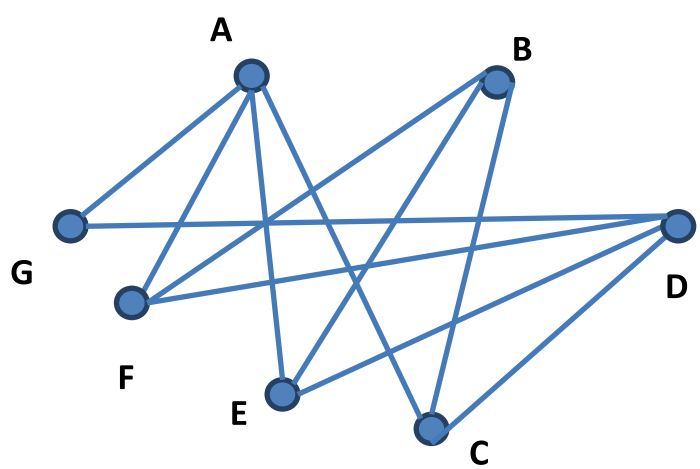


SOLUCIÓN:

(a) Grafo bipartito completo: AEBD | FC

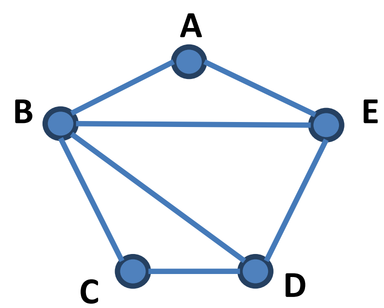


(b) Grafo bipartito (no completo) ABD | GFEC

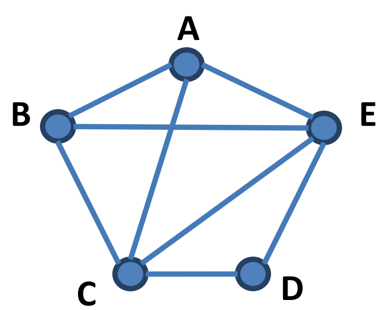


EJERCICIO 5: En caso de que haya alguno, indica para los siguientes grafos al menos uno de los circuitos y trayectorias eulerianos y uno de los circuitos y trayectorias hamiltonianos.

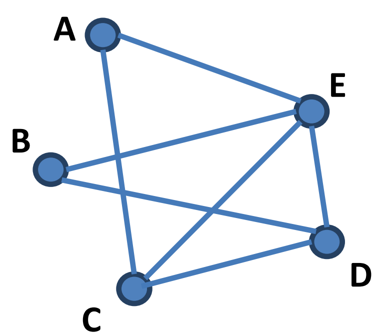
(a)



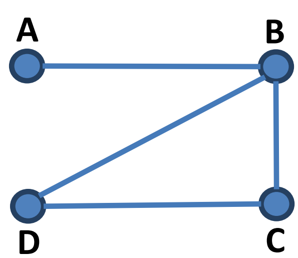
(b)



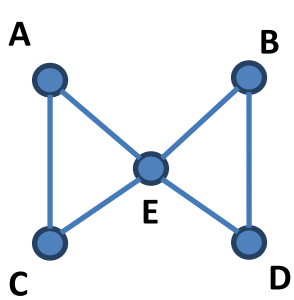
(c)



(d)

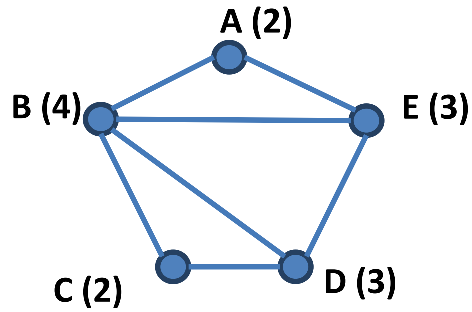


(e)

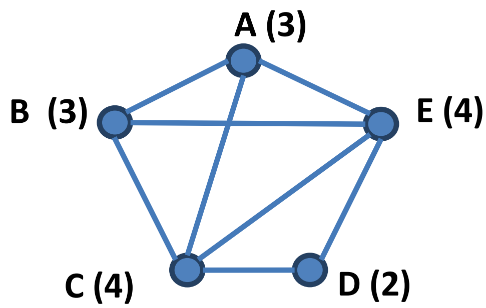


SOLUCIÓN:

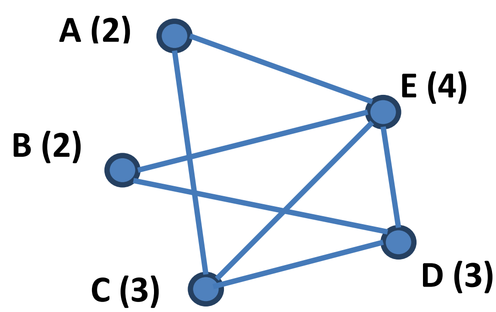
(a) Circuito euleriano: No hay
Trayectoria euleriana: EABEDBCD
Circuito hamiltoniano: ABCDEA
Trayectoria hamiltoniana: ABCDE



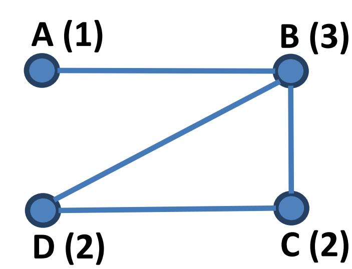
(b) Circuito euleriano: No hay
Trayectoria euleriana: ABCAEDCEB
Circuito hamiltoniano: ABCDEA
Trayectoria hamiltoniana: ABCDE



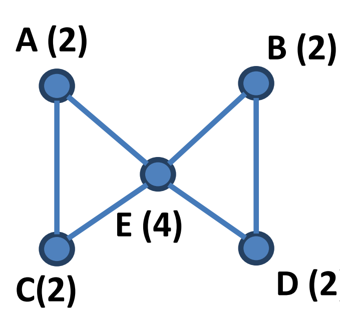
(c) Circuito euleriano: No hay
Trayectoria euleriana: DBEACEDC
Circuito hamiltoniano: ACDBEA
Trayectoria hamiltoniana: ACDBE



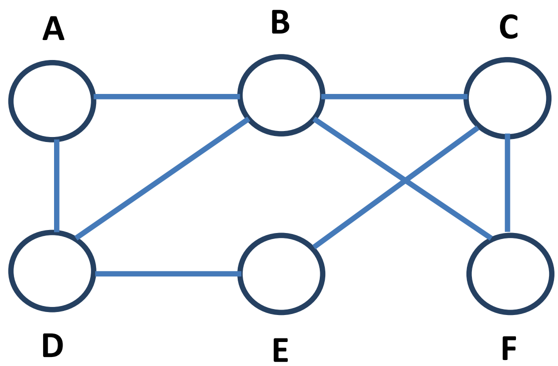
(d) Circuito euleriano: No hay
Trayectoria euleriana: ABDCB
Circuito hamiltoniano: No hay
Trayectoria hamiltoniana: ABDC



(e) Circuito euleriano: AEBDECA
Trayectoria euleriana: No hay
Circuito hamiltoniano: No hay
Trayectoria hamiltoniana: ACEBD



EJERCICIO 6: Considera el grafo



Encuentra el árbol abarcador desde el nodo A mediante

(a) BFS

(b) DFS

Indica paso a paso cómo se va generando el árbol.

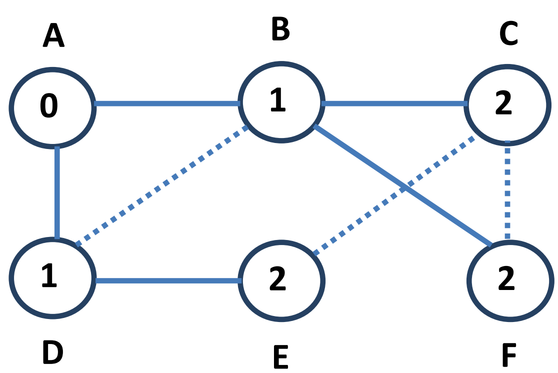
En BFS, indica el orden de exploración de los nodos (orden en que los nodos son extraídos de la cola).

En DFS etiqueta los nodos de acuerdo con el instante en que el nodo es visitado y el instante en que ha finalizado la exploración de dicho nodo.

En caso de que existan alternativas en el algoritmo, utiliza el orden alfabético de las etiquetas del nodo para decidir qué nodo es seleccionado.

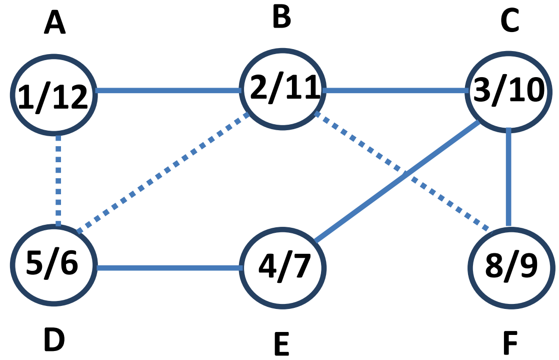
SOLUCIÓN:

BFS:

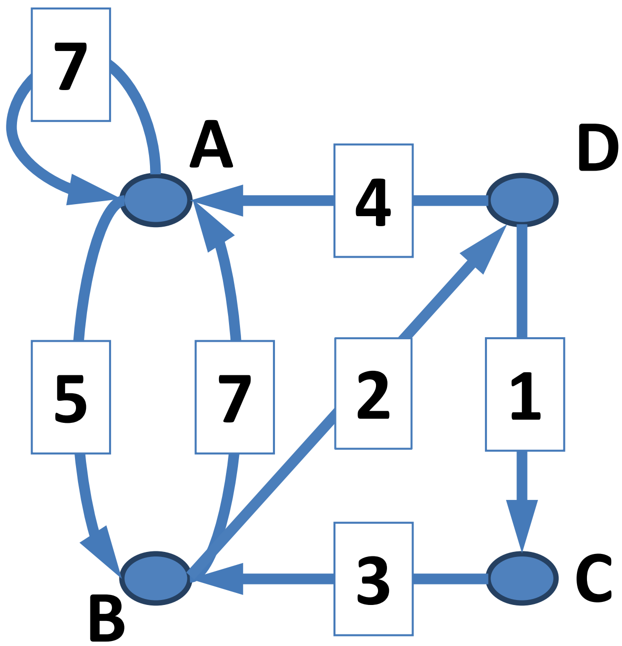


Orden de exploración: A, B, D, C, F, E

DFS:



EJERCICIO 7 [adaptado de Veerarajan, Cap. 7, Ejemplo 7.15]: Aplica el algoritmo de Warshall para encontrar los caminos más cortos (de menor peso) entre todos los pares de vértices del grafo dirigido que se muestra a continuación. El algoritmo debe ser modificado para que también encuentre los circuitos de longitud mínima no nula que comienzan y finalizan en cada uno de los vértices.



SOLUCIÓN:

W =

7	5	0	0
7	0	0	2
0	3	0	0
4	0	1	0

L_0 =

7_{AA}	5_{AB}	∞_{AC}	∞_{AD}
7_{BA}	∞_{BB}	∞_{BC}	2_{BD}
∞_{CA}	3_{CB}	∞_{CC}	∞_{CD}
4_{DA}	∞_{DB}	1_{DC}	∞_{DD}

L_1 =

7_{AA}	5_{AB}	∞_{AC}	∞_{AD}
7_{BA}	12_{BAB}	∞_{BC}	2_{BD}
∞_{CA}	3_{CB}	∞_{CC}	∞_{CD}
4_{DA}	9_{DAB}	1_{DC}	∞_{DD}

L_2 =

7_{AA}	5_{AB}	∞_{AC}	7_{ABD}
7_{BA}	12_{BAB}	∞_{BC}	2_{BD}
10_{CBA}	3_{CB}	∞_{CC}	5_{CBD}
4_{DA}	9_{DAB}	1_{DC}	11_{DABD}

L_3 =

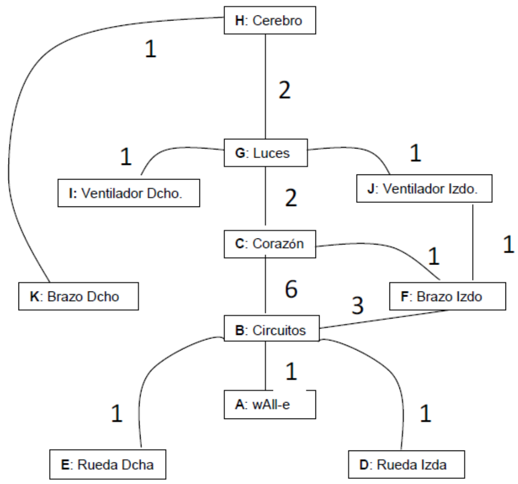
7_{AA}	5_{AB}	∞_{AC}	7_{ABD}
7_{BA}	12_{BAB}	∞_{BC}	2_{BD}
10_{CBA}	3_{CB}	∞_{CC}	5_{CBD}
4_{DA}	4_{DCB}	1_{DC}	6_{DCBD}

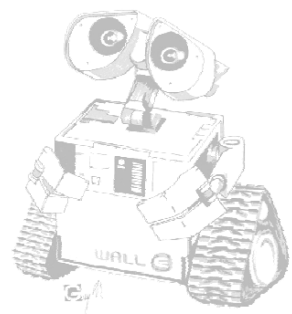
L_4 =

7_{AA}	5_{AB}	8_{ABDC}	7_{ABD}
6_{BDA}	6_{BDCB}	3_{BDC}	2_{BD}
9_{CBDA}	3_{CB}	6_{CBDC}	5_{CBD}
4_{DA}	4_{DCB}	1_{DC}	6_{DCBD}

GRAFO SIMPLIFICADO De LAS CONEXIONES DE wall-e

NOTA: En caso de que existan distintas alternativas en algún paso de los algoritmos implementados, debe utilizarse el orden alfabético. Para los algoritmos utiliza las etiquetas cortas de los nodos del grafo; esto es, el primer carácter que aparece en cada nodo: A, B, C, etc





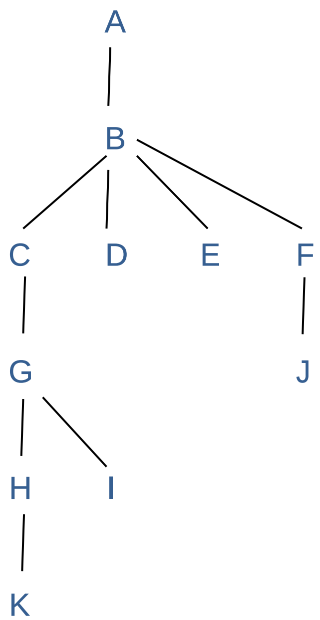
EJERCICIO 8: Dado el grafo simplificado de las conexiones de Wall-e, e ignorando pesos, genera el árbol de búsqueda en profundidad que parte de la placa con su nombre (nodo “A”). Detalla paso a paso el despliegue del árbol y etiqueta los nodos de acuerdo con el instante en que el nodo es visitado y el instante en que ha finalizado la exploración de dicho nodo.

SOLUCIÓN:

A(1/22), B(2/21), C(3/16), F(4/15), J(5/14), G(6/13), H(7/10), K(8/9), I(11/12), D(17/18), E(19/20)

EJERCICIO 9: Igualmente, dado el grafo simplificado de las conexiones de Wall-e, e ignorando pesos, genera el árbol de búsqueda en anchura que parte de la placa con su nombre (nodo "A"). Detalla paso a paso el despliegue del árbol.

SOLUCIÓN:



Orden de exploración: A-B-C-D-E-F-G-J-H-I-K

EJERCICIO 10: Dado el grafo simplificado de las conexiones de Wall-e, aplica el algoritmo de Dijkstra para encontrar el camino óptimo entre su cerebro (nodo H) y su rueda izquierda (nodo D). Indica el óptimo encontrado y la trayectoria a la que corresponde.
Ha de indicarse <u>si existe más de una trayectoria mínima posible, y por qué.</u>

SOLUCIÓN:

	L(0)	L(1)	L(2)	L(3)	L(4)	L(5)	L(6)	L(7)	L(8)	L(9)
A	∞	∞	∞	∞	∞	∞	∞	∞	$A^*_B(8)$	-
B	∞	∞	∞	∞	∞	∞	$B_C(10)$	$B^*_F(7)$	-	-
C	∞	∞	∞	C_G (4)	C_G (4)	C^*_G (4)	-	-	-	-
D	∞	∞	∞	∞	∞	∞	∞	∞	$D_B(8)$	$D^*_B(8)$
E	∞	∞	∞	∞	∞	∞	∞	∞	$E_B(8)$	$E_B(8)$
F	∞	∞	∞	∞	∞	$F_J(4)$	$F^*_J(4)$	-	-	-
G	∞	$G_H(2)$	$G^*_H(2)$	-	-	-	-	-	-	-
H	(0)	-	-	-	-	-	-	-	-	-
I	∞	∞	∞	I^*_G (3)	-	-	-	-	-	-
J	∞	∞	∞	J_G (3)	J^*_G (3)	-	-	-	-	-
K	∞	$K^*_H(1)$	-	-	-	-	-	-	-	-

La distancia más corta entre H y D es 8, y la trayectoria: H-G-J-F-B-D.
En L(8) se expandiría A, pero no tiene acceso con otros nodos, a continuación se procedería igual con D y con E, que se encuentran en la misma situación.

EJERCICIO 11: Se desea recablear a Wall-e, con coste mínimo, para que tenga más velocidad de reacción. Los costes de instalar las conexiones están especificados en los enlaces del grafo adjunto. El principio de diseño es que la red que se instale permita, en cada una de las etapas de instalación, que se comuniquen dos elementos cualesquiera ya conectados.
Explica el algoritmo a utilizar. Si tiene un nombre o relación con algún algoritmo impartido en clase indícalo. Dibuja la red que se obtiene si la instalación parte del cerebro (nodo H). Indica asimismo el orden en el que se añaden conexiones a la red ¿Cuál sería el coste de desplegar una red con estas características?

SOLUCIÓN:
Dado que la restricción es que en todo momento la red, aunque no abarque todo el grafo, debe ser conexa en cada una de las etapas del despliegue, el algoritmo a utilizar es Prim.

Aristas por orden creciente de peso:

	Arista analizada	Peso	Elegida/Descartada
arista 1	H-K	1	Elegida
arista 2	G-H	2	Elegida
arista 3	I-G	1	Elegida
arista 4	G-J	1	Elegida
arista 5	F-J	1	Elegida
arista 6	C-F	1	Elegida
arista 7	C-G	~~2~~	DESCARTADA x ciclo
arista 8	B-F	3	Elegida
arista 9	A-B	1	Elegida
arista 10	B-D	1	Elegida
arista 11	B-E	1	Elegida

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

- Incluir H-K (arista 1), se conectan H y K
- Incluir G-H (arista 2), se accede a G
- Incluir I-G (arista 3), se accede a I
- Incluir G-J (arista 4), se accede a J
- Incluir F-J (arista 5), se accede a F
- Incluir C-F (arista 6), se accede a C
- Descartar C-G (arista 7), porque se crearía un ciclo.
- Incluir B-F (arista 8), se accede a B
- Incluir A-B (arista 9), se accede a A
- Incluir B-D (arista 10), se accede a D
- Incluir B-E (arista 11), se accede a E

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 13.