

HOJA DE EJERCICIOS 3: Grafos y árboles EDyL 2022-2023

NOTA: Incluye explicaciones para tus respuestas. Un ejercicio cuya respuesta es correcta, pero que no incluye explicaciones podrá ser valorado como incompleto.

NOTA: En caso de que existan distintas alternativas en algún paso de los algoritmos implementados, debe utilizarse el orden alfabético con la convención de que los dígitos preceden a las letras.

EJERCICIO 1. Encuentra la trayectoria de coste mínimo en el **nodo a** y el **nodo i** utilizando el algoritmo de Dijkstra. Detalla cada paso del algoritmo en la tabla que se incluye a continuación, e indica cuál es la trayectoria óptima, así como su coste. Utiliza tantas filas y columnas de la tabla como sea necesario.

	L ₀	L ₁	L ₂	L ₃	L ₄	L ₅	L ₆	L ₇
A	(0A)	-	-	-	-	-	-	-
B	∞	(1A)	-	-	-	-	-	-
C	∞	∞	(4B)	-	-	-	-	-
D	∞	∞	∞	6C	6C	6C	(6C)	-
E	∞	∞	∞	∞	(5F)	-	-	-
F	∞	∞	4B	(4B)	-	-	-	-
G	∞	∞	∞	5C	5C	(5C)	-	-
H	∞	∞	∞	∞	∞	∞	7G	7G
I	∞	∞	∞	∞	6F	6F	6F	(6F)

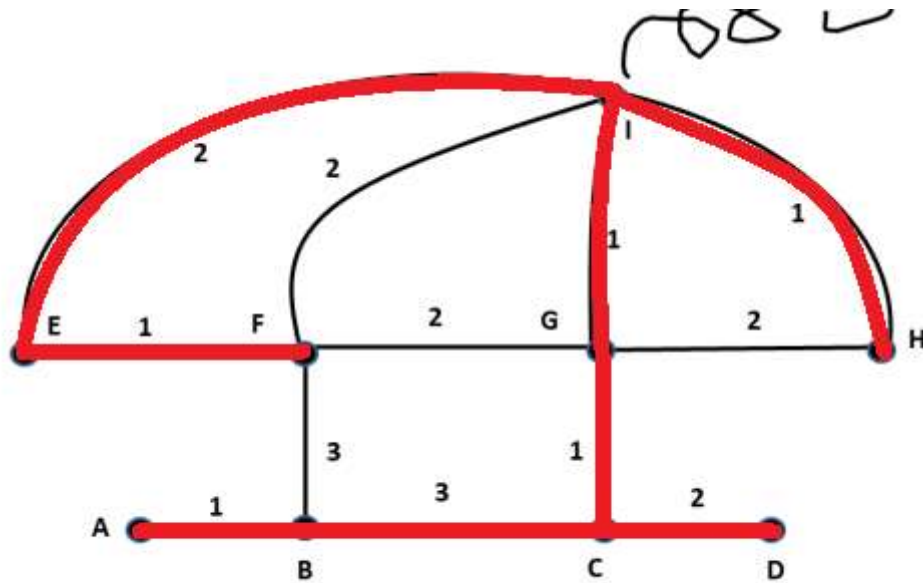
La trayectoria óptima es A-B-F-I, con coste 6.

EJERCICIO 2. Se desea realizar un procedimiento que permita acceder a todos los nodos del grafo al mínimo coste posible. Para ello, en cada paso del proceso se elegirá en tramo de menor coste.

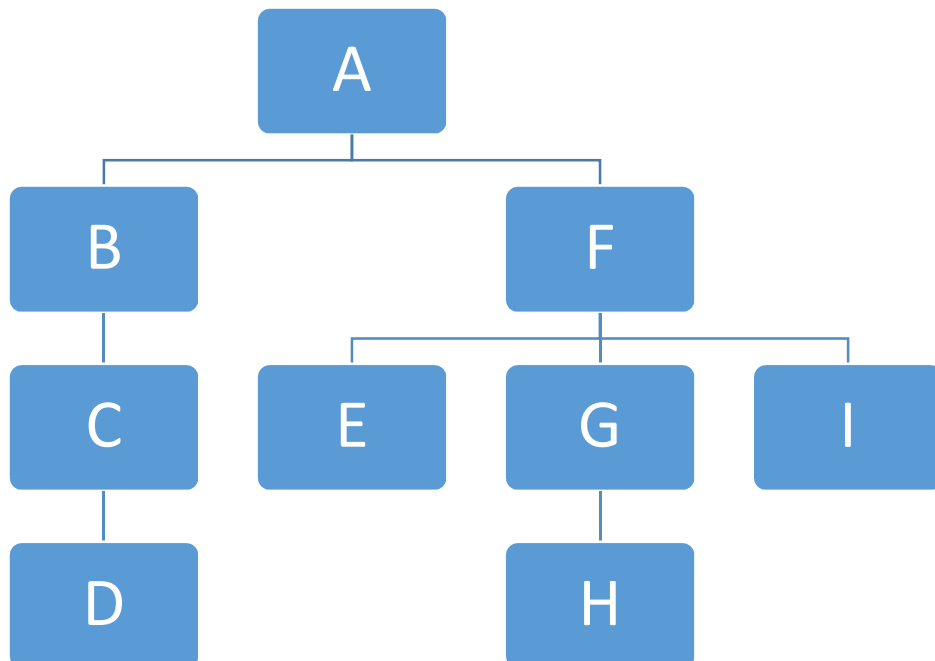
Se utilizará el orden alfabético si existen varias alternativas en algún paso de la aplicación del algoritmo utilizado. Utiliza tantas filas de la tabla incluida a continuación como sea necesario. Si alguna arista es examinada y descartada, indica cuál es la razón para ello. Si el algoritmo utilizado es conocido, indica un nombre. Representa el árbol resultante, así como su coste.

Arista examinada	Peso	Elegida/Descartada
A-B	1	Elegida
C-G	1	Elegida
E-F	1	Elegida
G-I	1	Elegida
H-I	1	Elegida
C-D	2	Elegida
E-I	2	Elegida
F-G	2	Descartada (circuito EFGI)
F-I	2	Descartada (circuito EFI)
G-H	2	Descartada (circuito GHI)
B-C	3	Elegida

Utilizando el algoritmo de Kruskal obtenemos un árbol de coste 12.



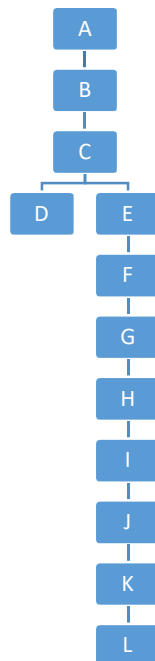
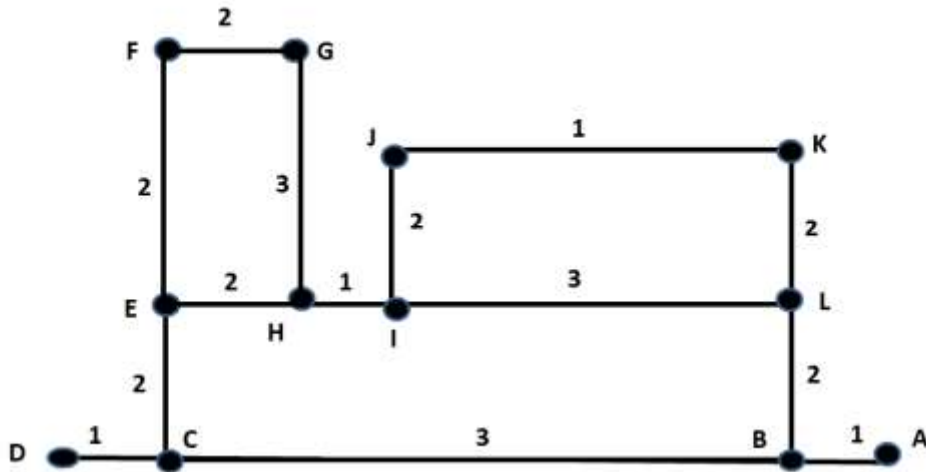
EJERCICIO 3. Por cuestiones de mantenimiento es necesario obtener el árbol de búsqueda en anchura (breadth-first search tree) correspondiente al grafo del que disponéis, comenzando en el **nodo a**. Representa el árbol obtenido y explica el procedimiento utilizado para obtenerlo.



1. Desde el nodo A exploramos los vértices adyacentes, primero el nodo B y después el nodo F. Cerramos el nodo A.
2. Desde el nodo B exploramos el único vértice adyacente, el nodo C. Cerramos el nodo B.
3. Desde el nodo F exploramos los vértices adyacentes, primero el E, después el G y por último el I. Cerramos el nodo F
4. Desde el nodo C exploramos el único vértice adyacente, el nodo D. Cerramos el nodo C.
5. Desde el nodo E no podemos explorar ningún vértice nuevo. Cerramos el nodo E.
6. Desde el nodo G exploramos el único vértice adyacente, el nodo H. Cerramos el nodo G.
7. Desde el nodo I no podemos explorar ningún vértice nuevo. Cerramos el nodo I.

8. Desde el nodo D no podemos explorar ningún vértice nuevo. Cerramos el nodo D.
9. Desde el nodo H no podemos explorar ningún vértice nuevo. Cerramos el nodo H.

EJERCICIO 4. Por cuestiones de mantenimiento es necesario obtener el árbol de búsqueda en profundidad (depth-first search tree) correspondiente al grafo del que disponéis, comenzando en el **nodo a**. Representa el árbol obtenido y explica el procedimiento utilizado para obtenerlo.



1. Desde el nodo A exploramos el nodo B.
2. Desde el nodo B exploramos el nodo C.
3. Desde el nodo C exploramos el nodo D.
4. Desde el nodo D no podemos explorar nodos nuevos. Cerramos el nodo D.
5. Desde el nodo C exploramos el nodo E.
6. Desde el nodo E exploramos el nodo F.
7. Desde el nodo F exploramos el nodo G.

8. Desde el nodo G exploramos el nodo H.
9. Desde el nodo H exploramos el nodo I.
10. Desde el nodo I exploramos el nodo J.
11. Desde el nodo J exploramos el nodo K.
12. Desde el nodo K exploramos el nodo L.
13. Desde el nodo L no podemos explorar nodos nuevos. Cerramos el nodo L. Cerramos el nodo K. Cerramos el nodo J. Cerramos el nodo I. Cerramos el nodo H. Cerramos el nodo G. Cerramos el nodo F. Cerramos el nodo E. Cerramos el nodo C. Cerramos el nodo B. Cerramos el nodo A.

EJERCICIO 5. Encuentra la trayectoria de coste mínimo entre el **nodo A** y el **nodo D** utilizando el algoritmo de Dijkstra. Detalla cada paso del algoritmo en la tabla que se incluye a continuación, e indica cuál es la trayectoria óptima, así como su coste. Utiliza tantas filas y columnas de la tabla como sea necesario.

	L ₀	L ₁	L ₂	L ₃	L ₄	L ₅	L ₆	L ₇
A	(0A)	-	-	-	-	-	-	-
B	∞	(2A)	-	-	-	-	-	-
C	∞	∞	(3B)	-	-	-	-	-
D	∞	∞	∞	8C	8C	8C	7G	(7G)
E	∞	∞	∞	∞	∞	∞	∞	∞
F	∞	∞	∞	∞	∞	∞	∞	∞
G	∞	∞	∞	4C	4C	(4C)	-	-
H	∞	∞	3B	(3B)	-	-	-	-
I	∞	3A	3A	3A	(3A)	-	-	-
J	∞	∞	∞	∞	∞	4I	(4I)	-

La trayectoria óptima es A-B-C-G-D, con coste 7.

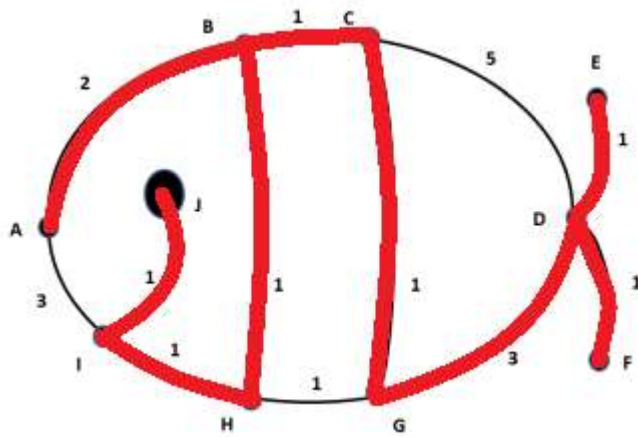
EJERCICIO 6. Se desea realizar un procedimiento que permita acceder a todos los nodos del grafo al mínimo coste posible. Para ello, en cada paso del proceso se elegirá el tramo de menor coste, garantizando la conectividad de los nodos ya elegidos durante el proceso.

Se utilizará orden alfabético si existen varias alternativas en algún paso de la aplicación del algoritmo utilizado. Utiliza tantas filas de la tabla incluida a continuación como sea necesario. Si alguna arista es examinada y descartada, indica cuál es la razón para ello. Si el algoritmo utilizado es conocido, indica su nombre. Representa el árbol resultante, así como su coste.

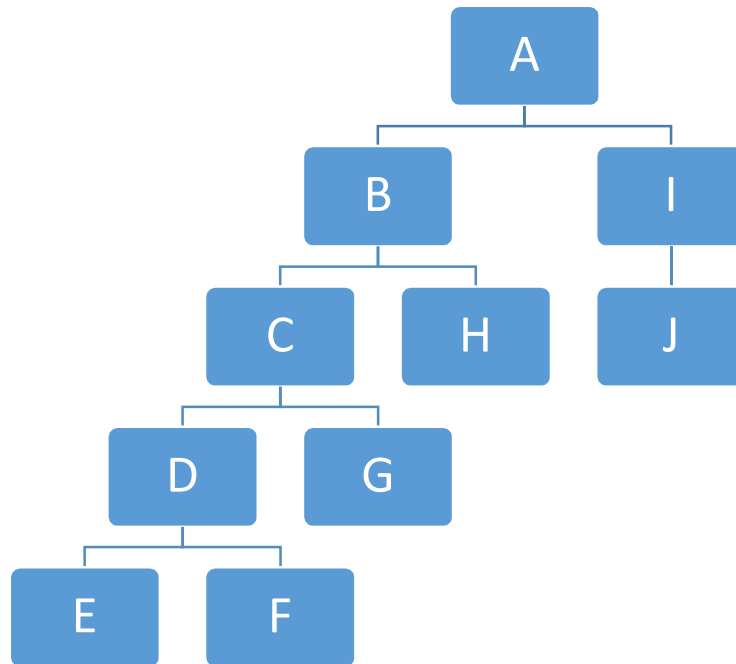
Nota: Se comenzará desde el **nodo A** sea cual sea el algoritmo utilizado.

Arista examinada	Peso	Elegida/Descartada
B-C	1	Elegida (2)
B-H	1	Elegida (3)
C-G	1	Elegida (4)
D-E	1	Elegida (8)
D-F	1	Elegida (9)
G-H	1	Descartada (circuito BCGH)
H-I	1	Elegida (5)
I-J	1	Elegida (6)
A-B	2	Elegida (1)
A-I	3	Descartada (circuito ABHI)
D-G	3	Elegida (7)

Utilizando el algoritmo de Prim obtenemos un árbol de coste 12.



EJERCICIO 7. Utiliza el algoritmo de búsqueda en anchura (BFS) para recorrer tu grafo, comenzando en el **nodo A**.



1. Desde el nodo A exploramos los vértices adyacentes, primero el nodo B y después el nodo I. Cerramos el nodo A.
2. Desde el nodo B exploramos los vértices adyacentes, primero el nodo C y después el nodo H. Cerramos el nodo B.
3. Desde el nodo I exploramos el único vértice adyacente, el nodo J. Cerramos el nodo I.
4. Desde el nodo C exploramos los vértices adyacentes, primero el nodo D y después el nodo G. Cerramos el nodo C.
5. Desde el nodo H no podemos explorar ningún vértice nuevo. Cerramos el nodo H.
6. Desde el nodo J no podemos explorar ningún vértice nuevo. Cerramos el nodo J.
7. Desde el nodo D exploramos los vértices adyacentes, primero el nodo E y después el nodo F. Cerramos el nodo D.
8. Desde el nodo G no podemos explorar ningún vértice nuevo. Cerramos el nodo G.

9. Desde el nodo E no podemos explorar ningún vértice nuevo. Cerramos el nodo E.
10. Desde el nodo F no podemos explorar ningún vértice nuevo. Cerramos el nodo F.