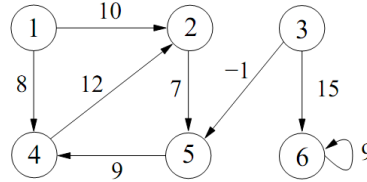


Ejercitación Teórica N° 4: **Grafos.**

Ejercicio 1.



(a) *Aplicando el recorrido DFS al grafo dirigido de la Figura 4, ¿cuáles son los vértices alcanzables desde el vértice 1 y en qué orden?*

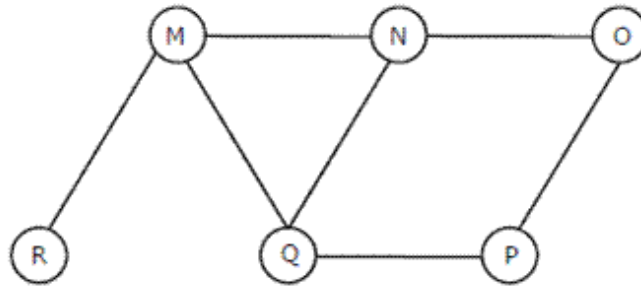
Aplicando el recorrido DFS al grafo dirigido de la Figura 4, los vértices alcanzables desde el vértice 1 son, en orden, 1, 2, 5 y 4.

(b) *Aplicando el recorrido BFS al grafo dirigido de la Figura 4, ¿cuáles son los vértices alcanzables desde el vértice 1 y en qué orden?*

Aplicando el recorrido BFS al grafo dirigido de la Figura 4, los vértices alcanzables desde el vértice 1 son, en orden, 1, 2, 4 y 5.

Ejercicio 2.

¿Cuál de los siguientes es un recorrido BFS válido para el grafo de la figura?



(a) *MNOPQR.*

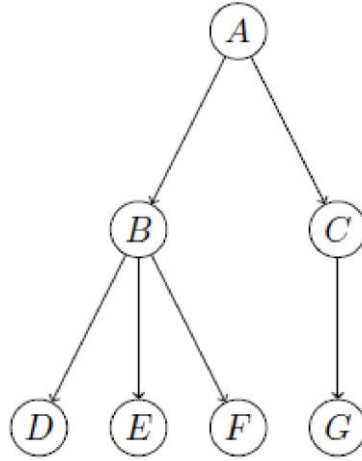
(b) *NQMPOR.*

(c) *QMNPOR.*

(d) *QMNPOR.*

Ejercicio 3.

*El siguiente árbol es el árbol que deriva de un recorrido BFS de un grafo dirigido G .
¿Cuál de las siguientes aristas no puede estar en G ?*



(a) (F, C) .

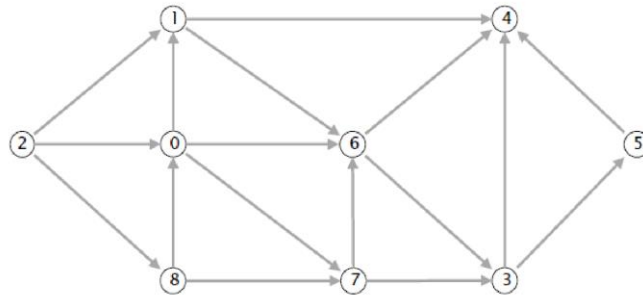
(b) (D, A) .

(c) (A, E) .

(d) (G, E) .

Ejercicio 4.

Se aplicó el recorrido DFS sobre el grafo dirigido de la siguiente figura, comenzando en el vértice 2. Asumir que las listas de adyacencias están ordenadas de menor a mayor.

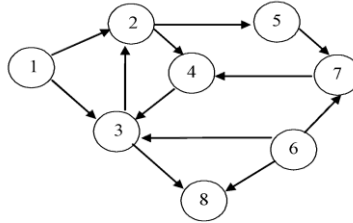


¿Cuál de las siguientes opciones corresponde al listado postorden de los vértices del grafo?

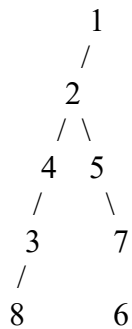
- (a) 2 0 6 4 3 5 7 1 8.
- (b) 4 5 3 6 7 0 1 8 2.
- (c) 4 5 3 6 1 7 0 8 2.
- (d) 2 0 1 8 6 7 4 3 5.

Ejercicio 5.

Dado el siguiente grado dirigido, en el siguiente bosque abarcador del DFS realizado a partir del vértice (1): 1, 2, 4, 3, 8, 5, 7, 6, habrá ...



- (a) 1 arco de cruce.
- (b) 2 arcos de cruce.
- (c) Más de 2 arcos de cruce.
- (d) Ninguna de las opciones.



Arcos del árbol: (1, 2), (2, 4), (4, 3), (3, 8), (2, 5), (5, 7).

Arcos de cruce: (7, 4), (6, 3), (6, 7), (6, 8).

Arcos de avance: (1, 3).

Arcos de retroceso: (3, 2).

Ejercicio 6.

Dado el grafo de la Figura 5, indicar cuál de las siguientes posibilidades es una ordenación topológica válida.

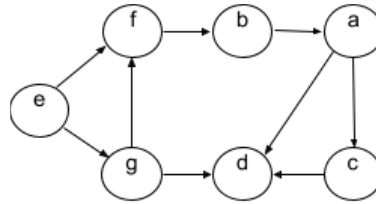


Figura 5

- (a) *e, g, d, f, b, a, c.*
- (b) *e, g, f, b, a, c, d.*
- (c) *Existe más de una posible ordenación topológica válida.*
- (d) *Ninguna de las otras respuestas es correcta.*

Aristas del grafo:

$e \rightarrow f$
 $e \rightarrow g$
 $f \rightarrow b$
 $g \rightarrow f$
 $g \rightarrow d$
 $b \rightarrow a$
 $a \rightarrow c$
 $a \rightarrow d$
 $c \rightarrow d$

Única ordenación topológica válida:

e, g, f, b, a, c, d.

Ejercicio 7.

Dado el grafo de la Figura 6, indicar cuál de las siguientes posibilidades es una ordenación topológica válida.

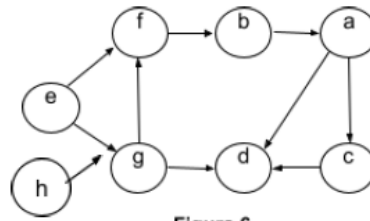


Figura 6

- (a) e, h, g, d, f, b, a, c.
- (b) e, g, f, b, a, c, d, h.
- (c) Existe más de una posible ordenación topológica válida.
- (d) Ninguna de las otras respuestas es correcta.

Aristas del grafo:

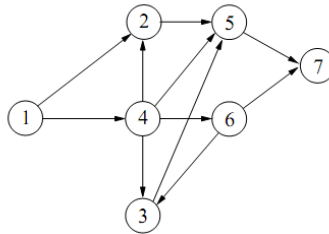
e → f
e → g
h → g
f → b
g → f
g → d
b → a
a → c
a → d
c → d

Única ordenación topológica válida:

e, h, g, f, b, a, c, d.

Ejercicio 8.

Aplicar las versiones de ordenación topológicas vistas en clase. 1 (usando arreglo), 2 (usando Cola o Pila) y 3 (usando DFS) del algoritmo que permite obtener la ordenación topológica del DAG de la Figura 7.



(a) Usando Arreglo.

Paso 1:

1 tiene 0 grado_in
 2 tiene 2 grado_in
 4 tiene 1 grado_in
 5 tiene 3 grado_in
 7 tiene 2 grado_in
 6 tiene 1 grado_in
 3 tiene 2 grado_in

Orden: [1].

Paso 2:

2 tiene 1 grado_in
 4 tiene 0 grado_in
 5 tiene 3 grado_in
 7 tiene 2 grado_in
 6 tiene 1 grado_in
 3 tiene 2 grado_in

Orden: [1, 4].

Paso 3:

2 tiene 0 grado_in
 5 tiene 2 grado_in
 7 tiene 2 grado_in
 6 tiene 0 grado_in
 3 tiene 1 grado_in

Orden: [1, 4, 2].

Paso 4:

5 tiene 1 grado_in
7 tiene 2 grado_in
6 tiene 0 grado_in
3 tiene 1 grado_in

Orden: [1, 4, 2, 6].

Paso 5:

5 tiene 1 grado_in
7 tiene 1 grado_in
3 tiene 0 grado_in

Orden: [1, 4, 2, 6, 3].

Paso 6:

5 tiene 0 grado_in
7 tiene 1 grado_in

Orden: [1, 4, 2, 6, 3, 5].

Paso 7:

7 tiene 0 grado_in

Orden: [1, 4, 2, 6, 3, 5, 7].

(b) *Usando Cola.*

Paso 1:

1 tiene 0 grado_in
2 tiene 2 grado_in
4 tiene 1 grado_in
5 tiene 3 grado_in
7 tiene 2 grado_in
6 tiene 1 grado_in
3 tiene 2 grado_in

Cola: [1].

Cola: [].

Orden: [1].

Paso 2:

2 tiene 1 grado_in
4 tiene 0 grado_in
5 tiene 3 grado_in
7 tiene 2 grado_in
6 tiene 1 grado_in
3 tiene 2 grado_in

Cola: [4].

Cola: [].

Orden: [1, 4].

Paso 3:

2 tiene 0 grado_in
5 tiene 2 grado_in
7 tiene 2 grado_in
6 tiene 0 grado_in
3 tiene 1 grado_in

Cola: [2, 6].

Cola: [6].

Orden: [1, 4, 2].

Paso 4:

5 tiene 1 grado_in
7 tiene 2 grado_in
3 tiene 1 grado_in

Cola: [6].

Cola: [].

Orden: [1, 4, 2, 6].

Paso 5:

5 tiene 1 grado_in
7 tiene 1 grado_in
3 tiene 0 grado_in

Cola: [3].

Cola: [].

Orden: [1, 4, 2, 6, 3].

Paso 6:

5 tiene 0 grado_in
7 tiene 1 grado_in

Cola: [5].

Cola: [].

Orden: [1, 4, 2, 6, 3, 5].

Paso 7:

7 tiene 0 grado_in

Cola: [7].

Cola: [].

Orden: [1, 4, 2, 6, 3, 5, 7].

(c) *Usando Pila.*

Paso 1:

1 tiene 0 grado_in

2 tiene 2 grado_in

4 tiene 1 grado_in

5 tiene 3 grado_in

7 tiene 2 grado_in

6 tiene 1 grado_in

3 tiene 2 grado_in

Pila: [1].

Pila: [].

Orden: [1].

Paso 2:

2 tiene 1 grado_in

4 tiene 0 grado_in

5 tiene 3 grado_in

7 tiene 2 grado_in

6 tiene 1 grado_in

3 tiene 2 grado_in

Pila: [4].

Pila: [].

Orden: [1, 4].

Paso 3:

2 tiene 0 grado_in

5 tiene 2 grado_in

7 tiene 2 grado_in

6 tiene 0 grado_in

3 tiene 1 grado_in

Pila: [2, 6].

Pila: [2].

Orden: [1, 4, 6].

Paso 4:

5 tiene 2 grado_in

7 tiene 1 grado_in

3 tiene 0 grado_in

Pila: [2, 3].

Pila: [2].

Orden: [1, 4, 6, 3].

Paso 5:

5 tiene 1 grado_in

7 tiene 1 grado_in

Pila: [2].

Pila: [].

Orden: [1, 4, 6, 3, 2].

Paso 6:

5 tiene 0 grado_in

7 tiene 1 grado_in

Pila: [5].

Pila: [].

Orden: [1, 4, 6, 3, 2, 5].

Paso 7:

7 tiene 0 grado_in

Pila: [7].

Pila: [].

Orden: [1, 4, 6, 3, 2, 5, 7].

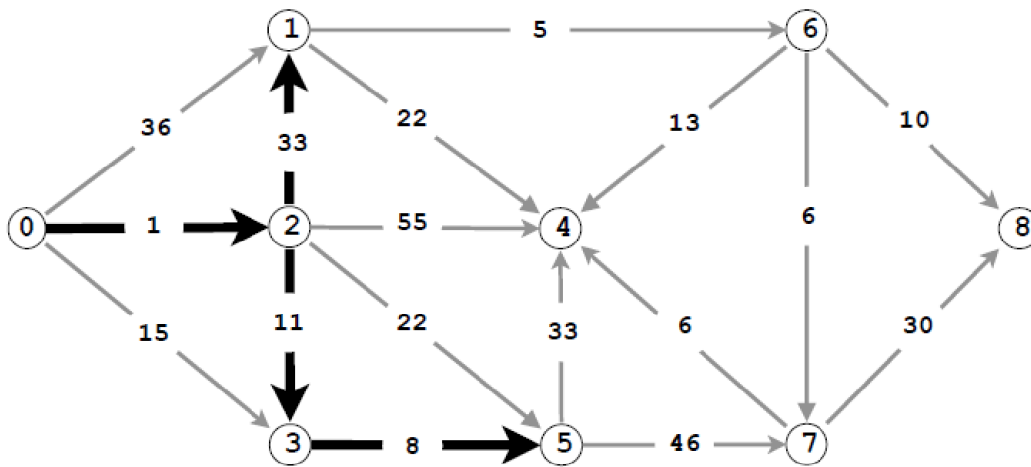
(d) Usando DFS.

Pila: [7, 5, 2, 3, 6, 4, 1].

Orden: [1, 4, 6, 3, 2, 5, 7].

Ejercicio 9.

Se ejecuta el algoritmo de Dijkstra sobre el siguiente digrafo pesado.



(a) La siguiente tabla contiene los valores luego de haberse procesado los vértices: 0, 2, 3, 5 y 1. Continuar la ejecución del algoritmo completando la tabla con los valores correspondientes.

(b) Completar la secuencia de vértices según el orden en el que el algoritmo de Dijkstra los toma (es decir, los considera "visitados"). Recordar que la ejecución del algoritmo comenzó por el vértice "0".

0 2 3 5 1

(c) Dibujar, sobre el grafo, los arcos (con trazo más grueso) del árbol abarcador resultante.

(d) Recuperar los vértices que componen los caminos de costo mínimo obtenidos con el algoritmo de Dijkstra, para los siguientes pares: (0, 5), (0, 7).

Ejercicio 10.

Ejercicio 11.

Ejercicio 12.

Ejercicio 13.

Ejercicio 14.

Ejercicio 15.