

- 1) Aplicando el recorrido DFS al grafo dirigido de la Figura 4, cuáles son los vértices alcanzables desde el vértice 1 y en qué orden.
 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.

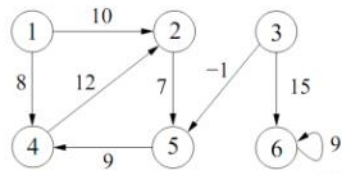
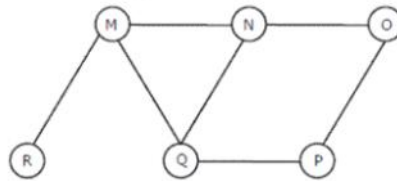


Figura 4

a. 1 2 5 4

b. 1 2 4 5

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



(a) MNOPQR

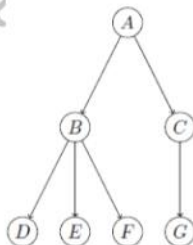
(b) NQMPOR

(c) QMNPRO

(d) QMNPOR

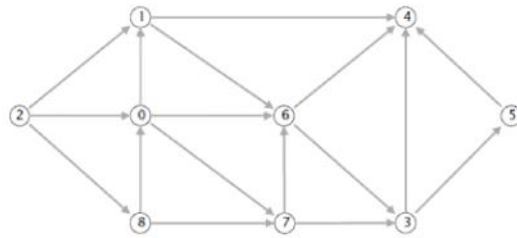
c y d

- 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?



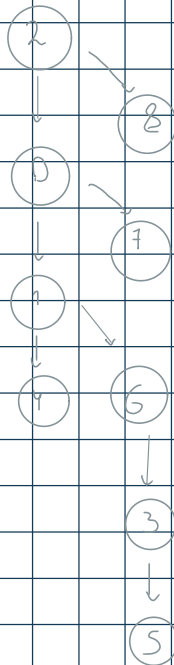
- i) (F,C)
- ii) (D,A)
- iii) (A,E)
- iv) (G,E)

- 4) Se aplicó el recorrido DFS sobre el grafo dirigido de la siguiente figura, comenzando en el vértice 2. Asuma 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?

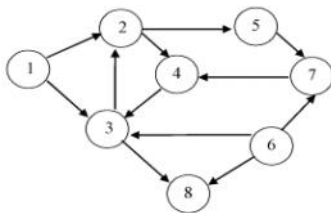
- (a) 2 0 6 4 3 5 7 1 8
- (b) 4 5 3 6 7 0 1 8 2
- (c) 4 5 3 6 1 7 0 8 2
- (d) 2 0 1 8 6 7 4 3 5



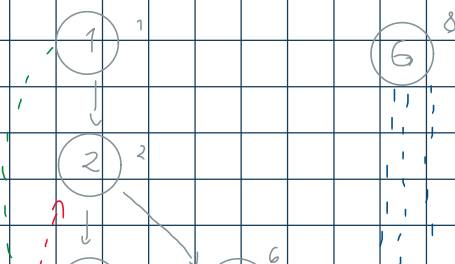
Respuesta correcta

Opción c

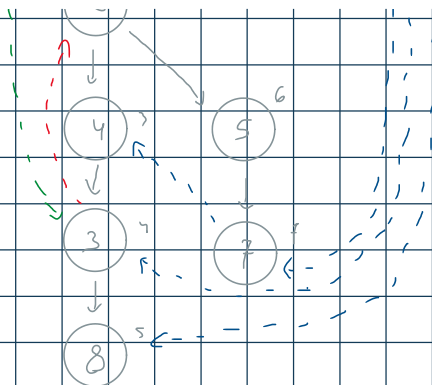
- 5) Dado el siguiente grafo dirigido, en el siguiente bosque abarcador del DFS realizado a partir del vértice (1): 1, 2, 4, 3, 8, 5, 7, 6, habrá ...



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



☒ avance
☒ retroceso
☒ cruce



☒ Retroceso

☐ Cruce

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

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

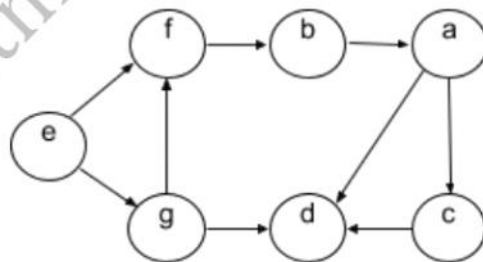


Figura 5

Respuesta iii

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

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

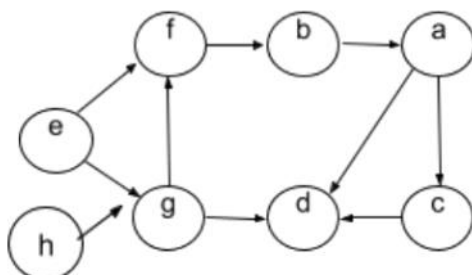


Figura 6

Respuesta vii

8) Aplique 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.

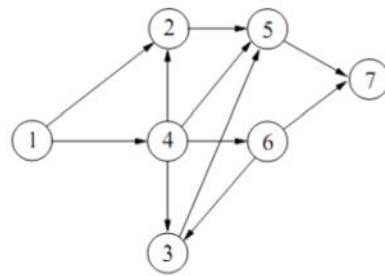


Figura 7

1 - Con arreglo

- Primero seleccionamos el vértice de grado-in 0 = ①

- Luego creamos un arreglo con los grados de entrada de cada vértice

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 2 | 2 | 1 | 3 | 1 | 2 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

Repetir proceso hasta

seleccionar todos los vértices

- Visitar vértice con grado-in 0, eliminarlo.

- Eliminar sus aristas salientes.

| | | | | | | |
|--------------|--------------|---|----------------|---|---|---|
| \checkmark | 1 | | $\checkmark 0$ | | | |
| 0 | 2 | 2 | 1 | 3 | 1 | 2 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

\checkmark = Visitado



Quiénes son los adyacentes de 4? 2, 5, 6, 3

| | | | | | | |
|--------------|----------------|--------------|----------------|--------------|--------------|---|
| \checkmark | $\checkmark 0$ | 1 | $\checkmark 0$ | 2 | 0 | |
| 0 | 2 | 2 | 1 | 3 | 1 | 2 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

1 2 3 4 5 6 7

(1) (4) (2)

Quiénes son los adyacentes de 2? 5

✓ ✓ ~~1~~ ✓ ~~2~~ ✓

| | | | | | | |
|---|---|--------------|---|--------------|---|---|
| 0 | 0 | 2 | 0 | 3 | 0 | 2 |
|---|---|--------------|---|--------------|---|---|

1 2 3 4 5 6 7

(1) (4) (2) (6)

Quiénes son los adyacentes de 6? 3, 7

✓ ✓ ~~0~~ ✓ ~~2~~ ✓ ~~1~~

| | | | | | | |
|---|---|--------------|---|--------------|---|--------------|
| 0 | 0 | 2 | 0 | 3 | 0 | 2 |
|---|---|--------------|---|--------------|---|--------------|

1 2 3 4 5 6 7

(1) (4) (2) (6) (3)

Quiénes son los adyacentes de 3? 5

✓ ✓ ✓ ✓ ~~0~~ ✓ ~~2~~ ~~1~~

| | | | | | | |
|---|---|---|---|--------------|---|--------------|
| 0 | 0 | 0 | 0 | 3 | 0 | 2 |
|---|---|---|---|--------------|---|--------------|

1 2 3 4 5 6 7

(1) (4) (2) (6) (3) (5)

Quiénes son los adyacentes de 5? 7

✓ ✓ ✓ ✓ ✓ ✓ ~~0~~ ~~1~~

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |



Con pila

Recorremos una vez el arreglo para

detectar vertices de grado $\text{in} = 0$

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 2 | 2 | 1 | 3 | 4 | 2 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

7
5
4
3
2
1
Pila

Si Tiene grado de entrada 0

lo apilo. Elimino sus aristas

Salientes / pregunta si el

$\text{grado}_{\text{in}} = 0$, si es igual lo

apilo. Cuando termino de

analizar sus aristas salientes,

desapilo. Repito el proceso



Con DFS

Se realiza un recorrido post orden, una vez

vistados todos los vertices a partir de uno

dato. Partiremos el dfs desde el vertice 1

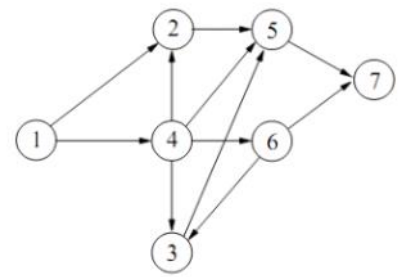
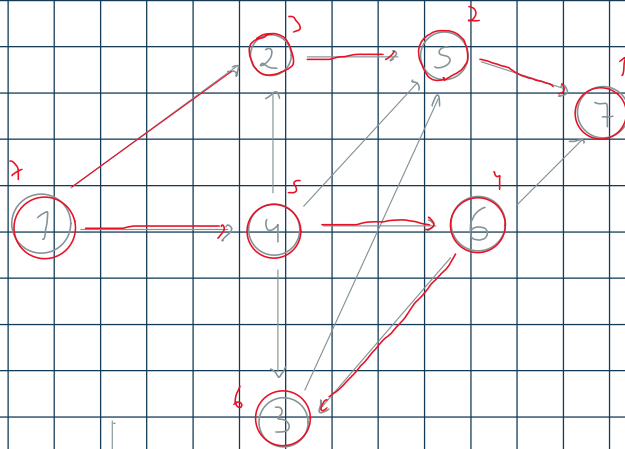


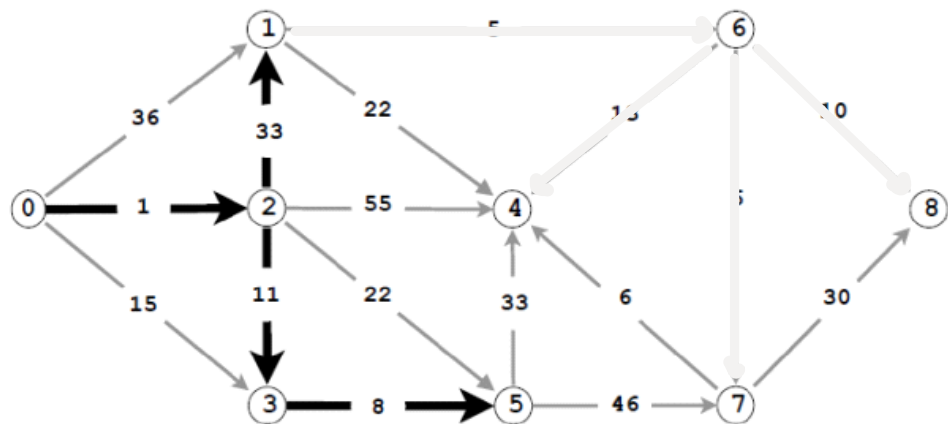
Figura 7

1
4
6
3
2
5
7

Sort Topologica: 1 4 6 3 2 5 7

Prta

9) Se ejecuta el algoritmo de **Dijkstra** sobre el siguiente dígrafo pesado.



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

| Nro. de iteración del algoritmo | Vértices | Distancia (0,v) | Vért. Previo | Visitado |
|---------------------------------|----------|-----------------|--------------|----------|
| 1° | 0 | 0 | - | 1 |
| 5° | 1 | ∞ 36 34 | 0 2 | 1 |
| 2° | 2 | ∞ 1 | 0 | 1 |
| 3° | 3 | ∞ 15 12 | 0 2 | 1 |

| | | | | |
|----|---|------------|-------|-----|
| 3° | 3 | ∞ 15 12 | 0 2 | 1 |
| 9° | 4 | ∞ 56 53 52 | 2 5 6 | 0 |
| 4° | 5 | ∞ 22 20 | 0 2 3 | 1 |
| 6° | 6 | ∞ 39 | 1 | 0 1 |
| 7° | 7 | ∞ 66 45 | 5 6 | 0 1 |
| 8° | 8 | ∞ 49 | 6 | 0 1 |

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

0 2 3 5 1 6 7 8 4

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

d) Recupere los vértices que componen los caminos de costo mínimo obtenidos con el algoritmo de Dijkstra, para los siguientes pares:

(0,5)

(0,7)

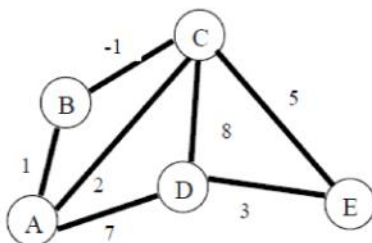
0,5 : 0, 1, 2, 3, 5

0,7 : 0, 2, 1, 6, 7

10) Dado el grafo pesado de la figura.

a.- ¿El algoritmo de Dijkstra funciona correctamente en este caso en particular, tomando como vértice origen a A?

b.- Si la respuesta es afirmativa, aplíquelo. Si la respuesta es negativa fundamente por qué no funciona el algoritmo.



| Nro | Vertices | distancia (0,v) | Vért. Provis | Visitado |
|-----|----------|--------------------|-----------------|----------|
| | A | 0 | - | 1 |
| | B | ∞ | 0 | 0 |
| | C | ∞ | 0 | 0 |
| | D | ∞ | 0 | 0 |
| | E | ∞ | 0 | 0 |

Selecciono el vertice A, actualizo B, C, D

| Nro | Vertices | distancia (0,v) | Vért. Provis | Visitado |
|-----|----------|--------------------|-----------------|----------|
| 1º | A | 0 | - | 1 |
| | B | 1 | A | 0 |
| | C | 1 | A | 0 |
| | D | 7 | A | 0 |
| | E | ∞ | 0 | 0 |

Visito vertice B y actualizo C

| Nro | Vertices | distancia (0,v) | Vért. Provis | Visitado |
|-----|----------|--------------------|-----------------|----------|
| 1º | A | 0 | - | 1 |
| 2º | B | 1 | A | 1 |
| | C | 0 | B | 0 |
| | D | 7 | A | 0 |
| | E | ∞ | 0 | 0 |

| | | | | |
|--|---|----------|---|---|
| | E | ∞ | 0 | 0 |
|--|---|----------|---|---|

Selecciona vertice C y actualizo E

| Nro | Vertices | distancia (0,v) | Vert previa | Visitado |
|-----|----------|--------------------|----------------|----------|
| 1º | A | 0 | - | 1 |
| 2º | B | 1 | A | 1 |
| 3º | C | 0 | B | 1 |
| | D | 7 | A | 0 |
| - | E | 5 | C | 0 |

Selecciona E y no actualizo nada

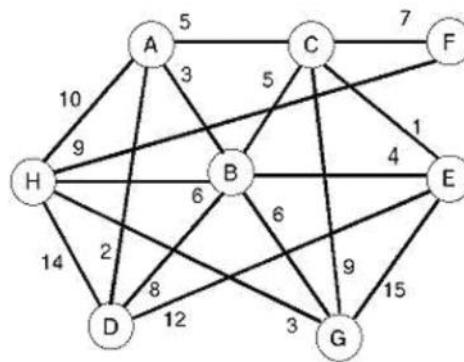
| Nro | Vertices | distancia (0,v) | Vert previa | Visitado |
|-----|----------|--------------------|----------------|----------|
| 1º | A | 0 | - | 1 |
| 2º | B | 1 | A | 1 |
| 3º | C | 0 | B | 1 |
| | D | 7 | A | 0 |
| 4º | E | 5 | C | 1 |

Selecciona D y no actualizo nada

| Nº | Vertices | distancia (0,v) | Vert. previo | Visitado |
|----|----------|--------------------|-----------------|----------|
| 1º | A | 0 | - | 1 |
| 2º | B | 1 | A | 1 |
| 3º | C | 0 | B | 1 |
| 5º | D | 7 | A | 1 |
| 4º | E | 5 | C | 1 |

Sí, funciona

11) Dado el siguiente grafo, ejecute el algoritmo de Dijkstra, partiendo del vértice H



b) ¿Cuáles fueron **los costos** intermedios encontrados por el algoritmo para encontrar el camino mínimo a E?

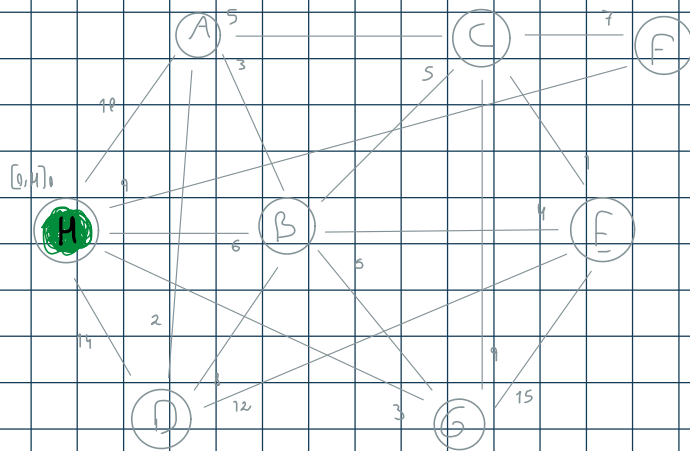
- 14, 11
- 18, 10
- 18, 11, 10
- 12, 11, 10
- Ninguna de las anteriores

c) ¿Cuáles fueron **los vértices** intermedios encontrados por el algoritmo para encontrar el camino mínimo a E?

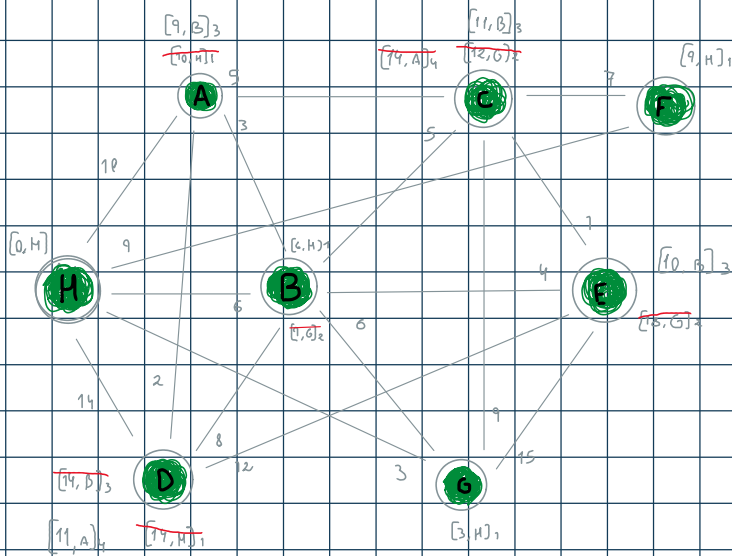
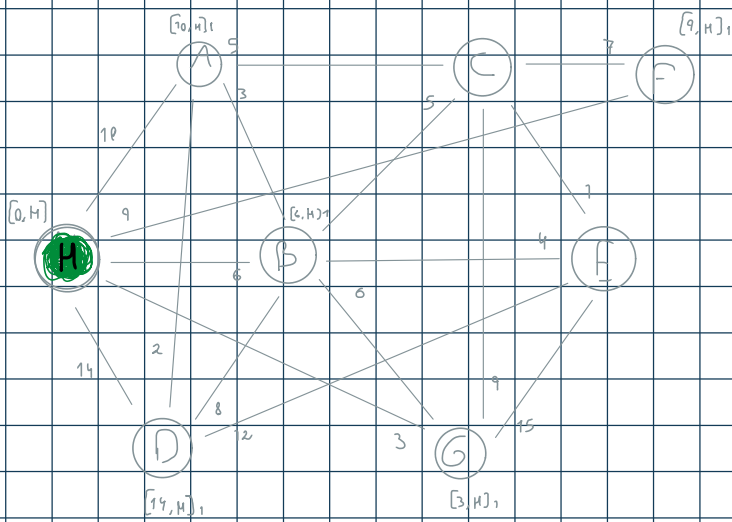
- A, C, B
- A, B, G
- G, B
- D, B
- A, C
- B

d) ¿En qué **iteración** del algoritmo fue tomado el vértice C?

- 4º
- 5º
- 6º
- 7º
- 8º



Verde = permanente

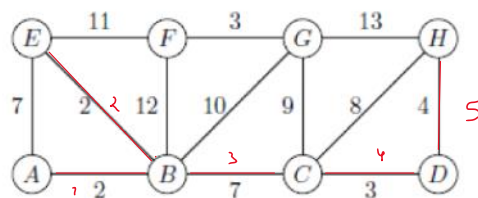


b 11, 10

c - G, B

d - 7°

12) Ejecute el algoritmo de Prim en el siguiente grafo, partiendo del vértice A. ¿Cuál es la suma de los pesos de la primera, tercera y quinta arista seleccionadas según el algoritmo?



(a) 9

(b) 10

(c) 11

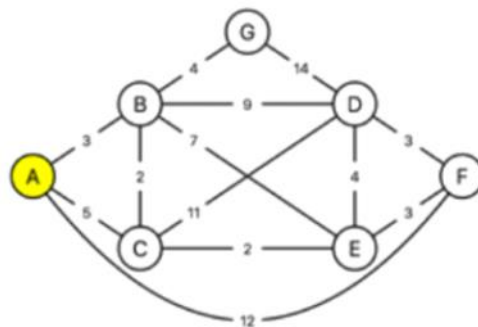
(d) 12

(e) 13

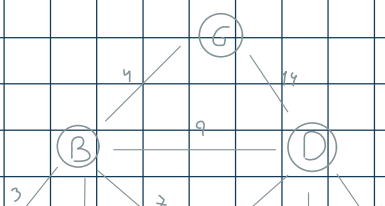
Opción d 13

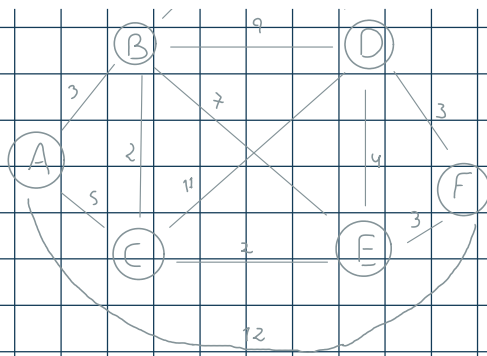
13) Obtener el árbol de expansión mínima utilizando el algoritmo de PRIM en el siguiente grafo comenzando del vértice A.

- Dibuje cómo evoluciona la construcción del árbol en cada paso.
- Muestre la ejecución del algoritmo en la tabla que aparece más abajo.
- Expresé el orden de ejecución del algoritmo (en cuanto a su eficiencia). Justifique su respuesta.

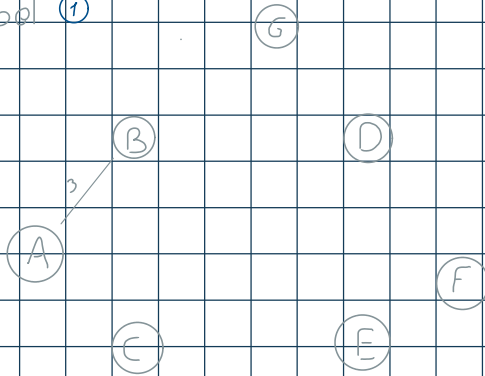


| Iteración en que se toma Arista (v,w) | Vértice "v" | Costo (v,w) | Vértice "w" | Visitado |
|---------------------------------------|-------------|-------------|-------------|----------|
| 1° | A | 0 | - | 0 1 |
| | B | ∞ | | 0 |
| | C | ∞ | | 0 |
| | D | ∞ | | 0 |
| | E | ∞ | | 0 |
| | F | ∞ | | 0 |
| | G | ∞ | | 0 |

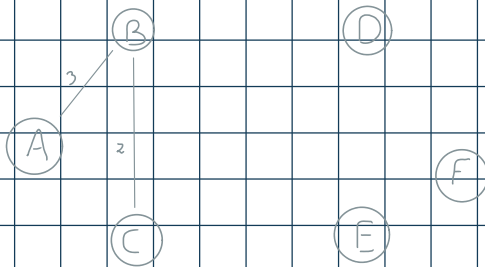




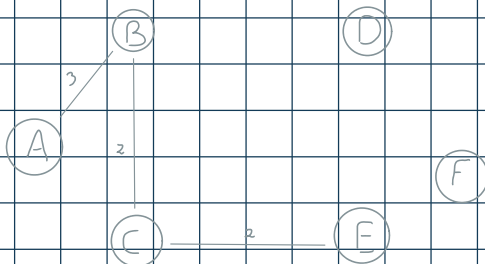
- árbol ①



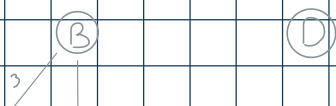
- árbol ②

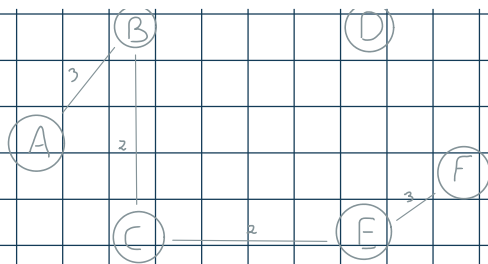


- árbol ③

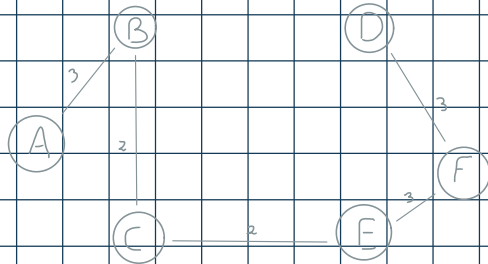


- árbol ④

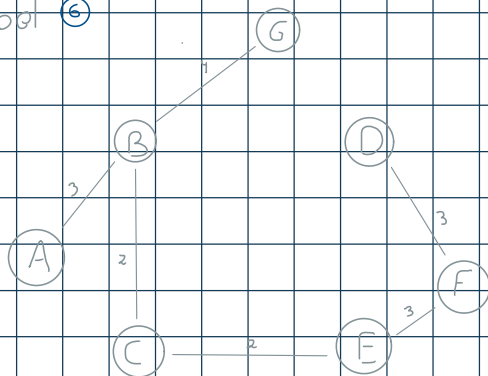




- árbol ⑤



- árbol ⑥



b.

| Iteración en que se toma Arista (v,w) | Vértice "v" | Costo (v,w) | Vértice "w" | Visitado |
|---------------------------------------|-------------|-------------|-------------|----------|
| 1° | A | 0 | - | 0 1 |
| 2° | B | ∞ | A | 0 1 |
| | C | ∞ | | 0 |
| | D | ∞ | | 0 |
| | E | ∞ | | 0 |
| | F | ∞ | | 0 |
| | G | ∞ | | 0 |

| Iteración en que se toma Arista (v,w) | Vértice "v" | Costo (v,w) | Vértice "w" | Visitado |
|---------------------------------------|-------------|-------------|-------------|----------|
| 1° | A | 0 | - | 0 1 |
| 2° | B | ∞ 3 | A | 0 1 |
| 3° | C | ∞ 1 | B | 0 1 |
| | D | ∞ | | 0 |
| | E | ∞ | | 0 |
| | F | ∞ | | 0 |
| | G | ∞ | | 0 |

| Iteración en que se toma Arista (v,w) | Vértice "v" | Costo (v,w) | Vértice "w" | Visitado |
|---------------------------------------|-------------|-------------|-------------|----------|
| 1° | A | 0 | - | 0 1 |
| 2° | B | ∞ 3 | A | 0 1 |
| 3° | C | ∞ 1 | B | 0 1 |
| | D | ∞ | | 0 |
| 4° | E | ∞ 2 | C | 0 1 |
| | F | ∞ | | 0 |
| | G | ∞ | | 0 |

| Iteración en que se toma Arista (v,w) | Vértice "v" | Costo (v,w) | Vértice "w" | Visitado |
|---------------------------------------|-------------|-------------|-------------|----------|
| 1° | A | 0 | - | 0 1 |
| 2° | B | ∞ 3 | A | 0 1 |
| 3° | C | ∞ 1 | B | 0 1 |
| | D | ∞ | | 0 |
| 4° | E | ∞ 2 | C | 0 1 |
| 5° | F | ∞ 3 | E | 0 1 |
| | G | ∞ | | 0 |

| Iteración en que se toma Arista (v,w) | Vértice "v" | Costo (v,w) | Vértice "w" | Visitado |
|---------------------------------------|-------------|-------------|-------------|----------|
| 1° | A | 0 | - | 0 1 |
| 2° | B | ∞ 3 | A | 0 1 |
| 3° | C | ∞ 1 | B | 0 1 |
| 6° | D | ∞ 3 | F | 0 |
| 4° | E | ∞ 2 | C | 0 1 |
| 5° | F | ∞ 3 | E | 0 1 |
| | G | ∞ | | 0 |

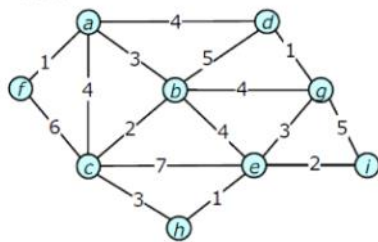
| Iteración en que se toma Arista (v,w) | Vértice "v" | Costo (v,w) | Vértice "w" | Visitado |
|---------------------------------------|-------------|-------------|-------------|----------|
| 1° | A | 0 | - | 0 1 |
| 2° | B | ∞ | A | 0 1 |
| 3° | C | ∞ | B | 0 1 |
| 6° | D | ∞ | F | 0 |
| 4° | E | ∞ | C | 0 1 |
| 5° | F | ∞ | E | 0 1 |
| 7° | G | ∞ | B | 0 1 |

a. Se hacen las mismas consideraciones que para el algoritmo de Dijkstra :

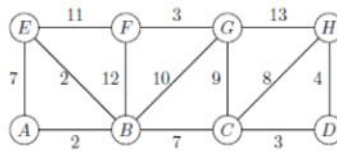
- Si se implementa con una tabla secuencial: El costo total del algoritmo es $O(|V|^2)$
 - Porque hay que recorrer V veces los vértices para sacar el mínimo y las actualizaciones se realizan E veces
- Si se implementa con heap: El costo total del algoritmo es $O(|E| \log |V|)$
 - Porque hay que hacer un deleteMin V veces que sería $\log |V|$ y las actualizaciones se realizan E veces (actualizar es de $\log V$, entonces sería $E * \log |V|$).

14) Obtener el árbol de expansión mínima utilizando el algoritmo de **Kruskal** en los siguientes grafos, dibujando cómo evoluciona la construcción del árbol en cada paso.

Expresa el orden de ejecución del algoritmo (en cuanto a su eficiencia). Justifique su respuesta.



Grafo 1



Grafo 2

e) Complete la secuencia de arcos del árbol abarcador mínimo, según el orden en que el algoritmo los incluye en el árbol

Grafo 1: _____

Grafo 2: _____

f) ¿Cuál es el costo del árbol abarcador resultante?

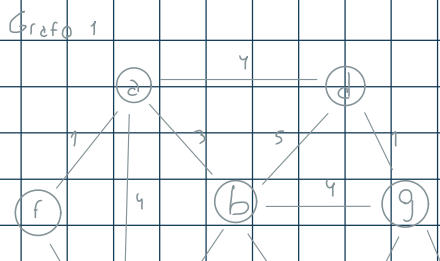
Grafo 1:

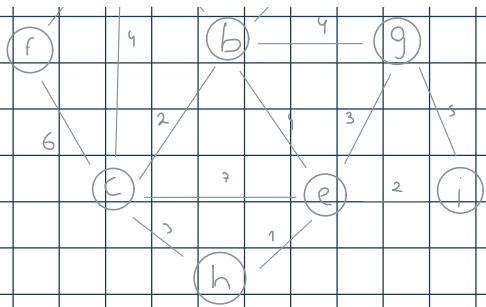
Grafo 2:

g) ¿Cuántos arcos fueron descartados durante el desarrollo del algoritmo hasta encontrar el árbol abarcador resultante?

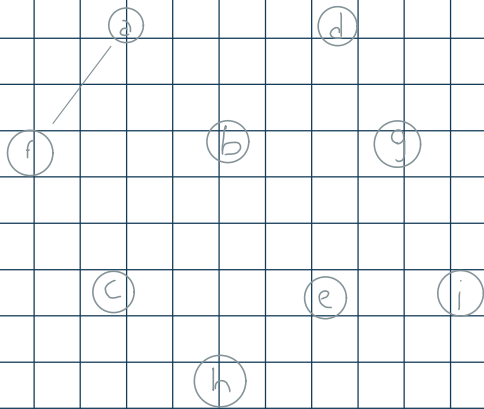
Grafo 1:

Grafo 2:

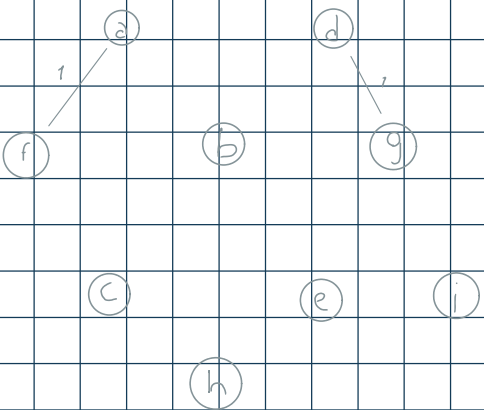




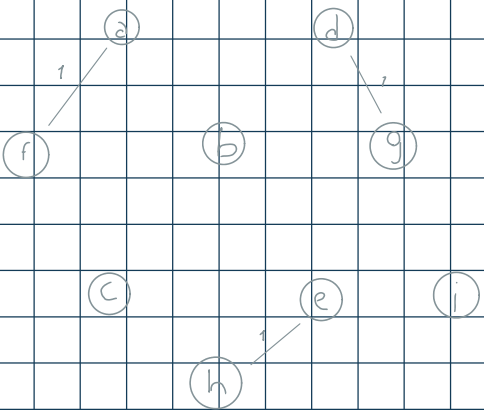
Arbol ①



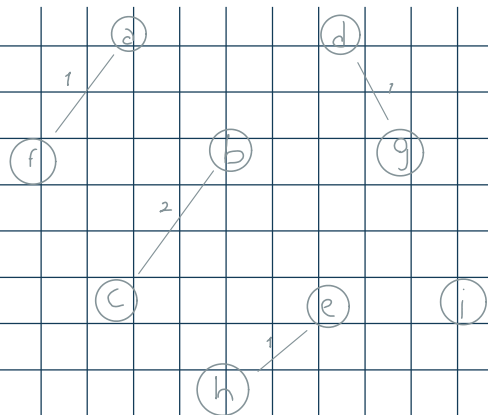
Arbol ②



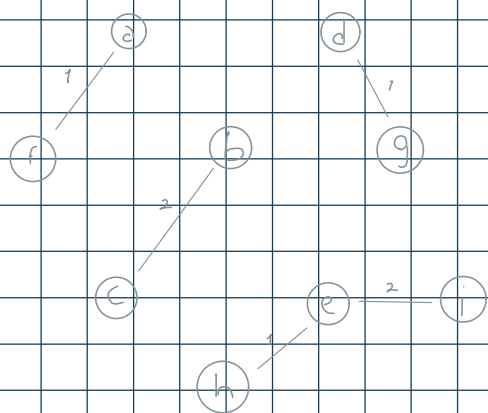
Arbol ③



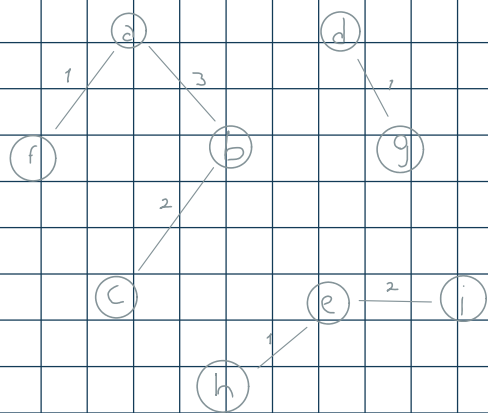
Arbol ④



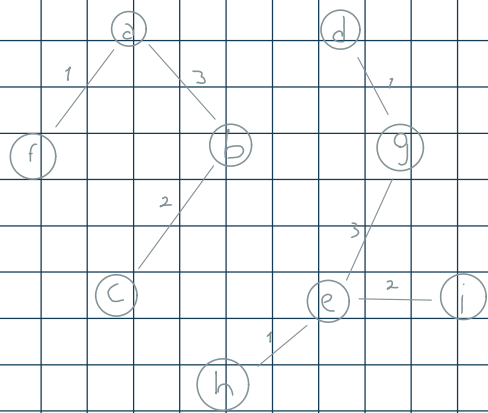
Arbol ⑤



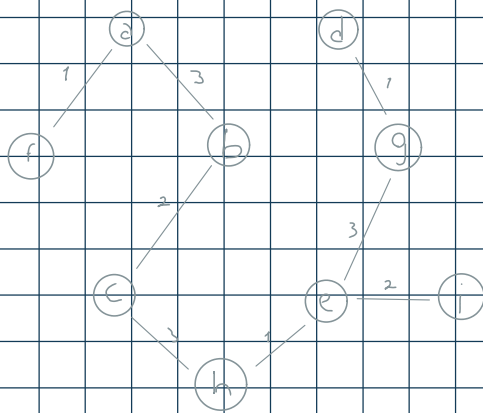
Arbol ⑥



Arbol ⑦



Arbol ⑧



1. $(a, f) \rightarrow 1$

2. $(d, g) \rightarrow 1$

3. $(h, e) \rightarrow 1$

4. $(c, h) \rightarrow 2$

5. $(e, i) \rightarrow 2$

6. $(a, b) \rightarrow 3$

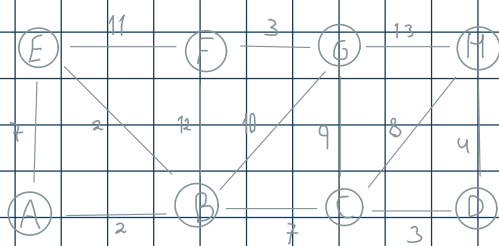
7. $(e, g) \rightarrow 3$

8. $(c, h) \rightarrow 3$

Costo 16

$\text{Arcos descartados} = \text{arcos totales} - (\text{Vertices} - 1) = 16 - 8 = 8$

Grafo 2



Arbol ①

(E) (F) (G) (H)

(A) — 2 — (B) (C) (D)

Arbol (2)

(E) — 2 — (F) (G) (H)

(A) — 2 — (B) (C) (D)

Arbol (3)

(E) — 2 — (F) — 3 — (G) (H)

(A) — 2 — (B) (C) (D)

Arbol (4)

(E) — 2 — (F) — 3 — (G) (H)

(A) — 2 — (B) (C) — 3 — (D)

Arbol (5)

(E) — 2 — (F) — 3 — (G) (H)

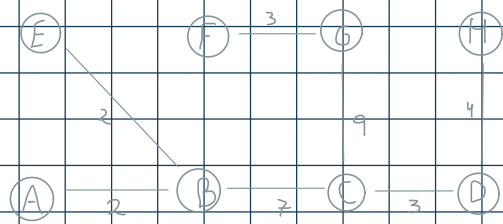
(A) — 2 — (B) (C) — 3 — (D)

Arbol (6)

(E) — 2 — (F) — 3 — (G) (H)

(A) — 2 — (B) — 7 — (C) — 3 — (D)

Arbol 7



1. $(a, b) \rightarrow 2$

2. $(e, b) \rightarrow 2$

3. $(f, g) \rightarrow 3$

4. $(c, d) \rightarrow 3$

5. $(h, d) \rightarrow 4$

6. $(b, c) \rightarrow 7$

7. $(g, d) \rightarrow 9$

Costo 30

Aristas faltantes 6.