Árboles y Grafos

Árbol de Cubrimiento Mínimo

Carlos Alberto Ramírez Restrepo

Programa de Ingeniería de Sistemas y Computación
Departamento de Electrónica y Ciencias de la Computación
Pontificia Universidad Javeriana
Cali, Colombia
carlosalbertoramirez@javerianacali.edu.co

Plan

Árbol de Cubrimiento Mínimo

Generalidades

- ✓ Sea $G = (V, E)$ un grafo conexo y no dirigido para el cuál se tiene una función de peso $w : E \to \mathbb{R}$ que asigna valores reales a las aristas.

Árbol de Cubrimiento Mínimo

Generalidades

- ✓ Sea $G = (V, E)$ un grafo conexo y no dirigido para el cuál se tiene una función de peso $w : E \rightarrow \mathbb{R}$ que asigna valores reales a las aristas.
- ✓ Se dice que $T = (V_T, E_T)$ es un árbol de cubrimiento de G sii: i) T es un árbol, ii) $V_T = V$ y iii) $E_T \subseteq E$.
- ✓ Un árbol de cubrimiento corresponde a cualquier subgrafo de G que es un árbol, contiene algunas aristas de G y contiene todos los vértices en G.

Árbol de Cubrimiento Mínimo

Generalidades

- ✓ El peso de un árbol de cubrimiento corresponde a la suma de sus aristas.

Árbol de Cubrimiento Mínimo

Generalidades

- ✓ El peso de un árbol de cubrimiento corresponde a la suma de sus aristas.
- ✓ Un árbol de cubrimiento mínimo (*minimum spanning tree*) es un árbol de cubrimiento cuyo peso es mínimo.

Árbol de Cubrimiento Mínimo

Generalidades

- ✓ El peso de un árbol de cubrimiento corresponde a la suma de sus aristas.
- ✓ Un árbol de cubrimiento mínimo (*minimum spanning tree*) es un árbol de cubrimiento cuyo peso es mínimo.
- ✓ Cuando el grafo G no es conexo, se habla de un bosque de cubrimiento mínimo (*minimum spanning forest*).

Árbol de Cubrimiento Mínimo

Generalidades

- ✓ El peso de un árbol de cubrimiento corresponde a la suma de sus aristas.
- ✓ Un árbol de cubrimiento mínimo (*minimum spanning tree*) es un árbol de cubrimiento cuyo peso es mínimo.
- ✓ Cuando el grafo G no es conexo, se habla de un bosque de cubrimiento mínimo (*minimum spanning forest*).
- ✓ Un bosque de cubrimiento mínimo en un grafo no conexo corresponde a la unión de los árboles de cubrimiento mínimo de cada componente.

Árbol de Cubrimiento Mínimo

Generalidades

- ✓ Si $|V| = n$ entonces el árbol de cubrimiento mínimo tiene $n - 1$ aristas.

Árbol de Cubrimiento Mínimo

Generalidades

- ✓ Si $|V| = n$ entonces el árbol de cubrimiento mínimo tiene $n - 1$ aristas.
- ✓ Un grafo puede tener más de un árbol de cubrimiento mínimo aunque si todas sus aristas tienen peso diferente solo puede haber uno.

Árbol de Cubrimiento Mínimo

Generalidades

- ✓ Si $|V| = n$ entonces el árbol de cubrimiento mínimo tiene $n-1$ aristas.
- ✓ Un grafo puede tener más de un árbol de cubrimiento mínimo aunque si todas sus aristas tienen peso diferente solo puede haber uno.
- ✓ Si el peso de todas las aristas es igual entonces cualquier árbol de cubrimiento es un árbol de cubrimiento mínimo.

Árbol de Cubrimiento Mínimo

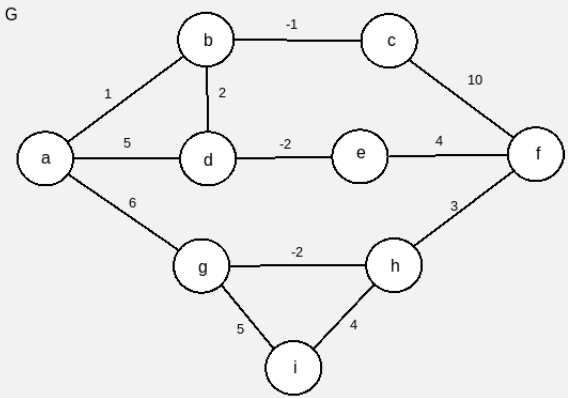
Especificación Problema

- ✓ **Entrada**: Un grafo $G = (V, E)$ conexo y no dirigido con función de peso $w : E \rightarrow \mathbb{R}$.
- ✓ **Salida**: $E_T \subseteq E$ tal que (V, E_T) es un árbol de cubrimiento mínimo de G.

Árbol de Cubrimiento Mínimo

Ejemplo

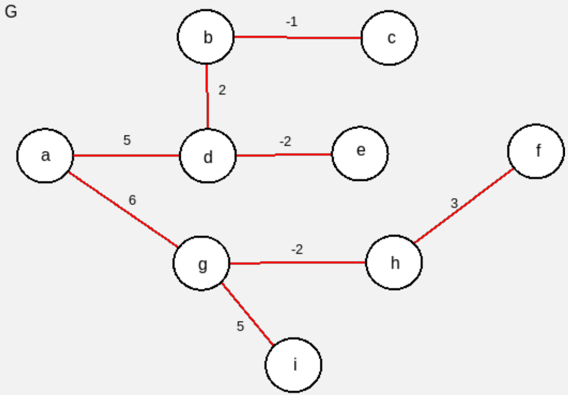
Considere el siguiente grafo;



Árbol de Cubrimiento Mínimo

Ejemplo

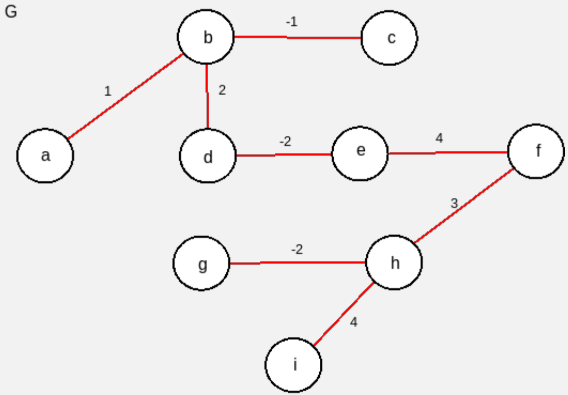
Un árbol de cubrimiento del grafo puede ser;



Árbol de Cubrimiento Mínimo

Ejemplo

El árbol de cubrimiento mínimo es el siguiente;



Árbol de Cubrimiento Mínimo

Generalidades

- ✓ La idea general para construir un MST de $G = (V, E, w: \ E \to \mathbb{R})$ usando la técnica de diseño voraz es mantener un conjunto de aristas $A \subseteq E$ que cumpla con el siguiente invariante:

$$A \text{ hace parte de las aristas de un MST de } G$$

Árbol de Cubrimiento Mínimo

Generalidades

- La idea general para construir un MST de $G = (V, E, w : \ E \to \mathbb{R})$ usando la técnica de diseño voraz es mantener un conjunto de aristas $A \subseteq E$ que cumpla con el siguiente invariante:

  A hace parte de las aristas de un MST de G

- De esta manera, se debe construir incrementalmente A, agregando aristas mientras se respeta el invariante.

Árbol de Cubrimiento Mínimo

Generalidades

- ✓ La idea general para construir un MST de $G = (V, E, w : E \to \mathbb{R})$ usando la técnica de diseño voraz es mantener un conjunto de aristas $A \subseteq E$ que cumpla con el siguiente invariante:

$$A \text{ hace parte de las aristas de un MST de } G$$

- ✓ De esta manera, se debe construir incrementalmente A, agregando aristas mientras se respeta el invariante.
- ✓ Cuando no sea posible agregar aristas, A incluirá únicamente aristas de un MST de G.

Árbol de Cubrimiento Mínimo

Generalidades

El algoritmo general para construir el MST tendrá la siguiente estructura:

```
Generic-MST(G = (V, E, w)):
1. A = empty
2. Mientras A no sea un árbol de cubrimiento de G
3.   Encontrar una arista segura e en E para extender A
4.   A = A union {e}
5. Retornar A
```

Árbol de Cubrimiento Mínimo

Generalidades

- ✓ A partir de lo anterior es necesario determinar cuándo una arista es segura para extender A.

Árbol de Cubrimiento Mínimo

Generalidades

- ✓ A partir de lo anterior es necesario determinar cuándo una arista es segura para extender A.
- ✓ Un corte de V es una pareja (S_0, S_1) tal que $\{S_0, S_1\}$ es una partición de V.

Árbol de Cubrimiento Mínimo

Generalidades

- ✓ A partir de lo anterior es necesario determinar cuándo una arista es segura para extender A.
- ✓ Un corte de V es una pareja (S_0, S_1) tal que $\{S_0, S_1\}$ es una partición de V.
- ✓ Un arco $e \in E$ cruza un corte (S_0, S_1) de V sii uno de los vértices de e está en S_0 y el otro está en S_1.

Árbol de Cubrimiento Mínimo

Generalidades

- ✓ A partir de lo anterior es necesario determinar cuándo una arista es segura para extender A.
- ✓ Un corte de V es una pareja (S_0, S_1) tal que $\{S_0, S_1\}$ es una partición de V.
- ✓ Un arco $e \in E$ cruza un corte (S_0, S_1) de V sii uno de los vértices de e está en S_0 y el otro está en S_1.
- ✓ Un corte (S_0, S_1) de V respeta un subconjunto $E' \subseteq E$ si ningún arco en E' cruza (S_0, S_1).

Árbol de Cubrimiento Mínimo

Generalidades

- ✓ A partir de lo anterior es necesario determinar cuándo una arista es segura para extender A.
- ✓ Un corte de V es una pareja (S_0, S_1) tal que $\{S_0, S_1\}$ es una partición de V.
- ✓ Un arco $e \in E$ cruza un corte (S_0, S_1) de V sii uno de los vértices de e está en S_0 y el otro está en S_1.
- ✓ Un corte (S_0, S_1) de V respeta un subconjunto $E' \subseteq E$ si ningún arco en E' cruza (S_0, S_1).
- ✓ Un arco $e \in E$ es ligero para un corte (S_0, S_1) de V sii $w(e)$ es mínimo entre todos los arcos que cruzan (S_0, S_1).

Árbol de Cubrimiento Mínimo

Generalidades

Teorema: Sea $G = (V, E, w: \ E \rightarrow \mathbb{R})$ un grafo conexo y no dirigido. Sea $A \subseteq E$ de aristas que hacen parte de algún MST de G. Sea $(S, V \backslash S)$ un corte de V que respeta A. Si $(u, v) \in E$ es un arco ligero que cruza $(S, V \backslash S)$ entonces (u, v) es seguro para A.

Árbol de Cubrimiento Mínimo

Generalidades

Teorema: Sea $G = (V, E, w: \ E \to \mathbb{R})$ un grafo conexo y no dirigido. Sea $A \subseteq E$ de aristas que hacen parte de algún MST de G. Sea $(S, V \backslash S)$ un corte de V que respeta A. Si $(u, v) \in E$ es un arco ligero que cruza $(S, V \backslash S)$ entonces (u, v) es seguro para A.

Demostración: Sea $T = (V, E_T)$ un MST tal que $A \subseteq E_T$. Se procede por casos:

- ✓ $(u, v) \in E_T$
- ✓ $(u, v) \notin E_T$

Árbol de Cubrimiento Mínimo

Generalidades

Demostración: Sea $T = (V, E_T)$ un MST tal que $A \subseteq E_T$. Se procede por casos:

- ✓ Si $(u, v) \in E_T$ entonces se tiene que (u, v) es seguro para A.

Árbol de Cubrimiento Mínimo

Generalidades

Demostración: Sea $T = (V, E_T)$ un MST tal que $A \subseteq E_T$. Se procede por casos:

- ✓ Si $(u, v) \in E_T$ entonces se tiene que (u, v) es seguro para A.
- ✓ Si $(u, v) \notin E_T$ entonces (u, v) forma un ciclo con las aristas del camino simple entre u y v en T.

Árbol de Cubrimiento Mínimo

Generalidades

Demostración: Sea $T = (V, E_T)$ un MST tal que $A \subseteq E_T$. Se procede por casos:

- ✓ Si $(u, v) \in E_T$ entonces se tiene que (u, v) es seguro para A.
- ✓ Si $(u, v) \notin E_T$ entonces (u, v) forma un ciclo con las aristas del camino simple entre u y v en T. Dado que u y v están en lados opuestos del corte $(S, V \backslash S)$ al menos una arista de T hace parte del camino entre u y v y cruza el corte $(S, V \backslash S)$.

Árbol de Cubrimiento Mínimo

Generalidades

Demostración: Sea $T = (V, E_T)$ un MST tal que $A \subseteq E_T$. Se procede por casos:

- ✓ Si $(u, v) \in E_T$ entonces se tiene que (u, v) es seguro para A.
- ✓ Si $(u, v) \notin E_T$ entonces (u, v) forma un ciclo con las aristas del camino simple entre u y v en T. Dado que u y v están en lados opuestos del corte $(S, V \backslash S)$ al menos una arista de T hace parte del camino entre u y v y cruza el corte $(S, V \backslash S)$.

  Sea (x, y) dicha arista. (x, y) no está en A puesto que el corte $(S, V \backslash S)$ respeta A.

Árbol de Cubrimiento Mínimo

Generalidades

Demostración: Sea $T = (V, E_T)$ un MST tal que $A \subseteq E_T$. Se procede por casos:

- ✓ Si $(u, v) \in E_T$ entonces se tiene que (u, v) es seguro para A.
- ✓ Si $(u, v) \notin E_T$ entonces (u, v) forma un ciclo con las aristas del camino simple entre u y v en T. Dado que u y v están en lados opuestos del corte $(S, V \backslash S)$ al menos una arista de T hace parte del camino entre u y v y cruza el corte $(S, V \backslash S)$.

  Sea (x, y) dicha arista. (x, y) no está en A puesto que el corte $(S, V \backslash S)$ respeta A. Si se elimina (x, y) de T, este se divide en dos componentes.

Árbol de Cubrimiento Mínimo

Generalidades

Demostración: Sea $T = (V, E_T)$ un MST tal que $A \subseteq E_T$. Se procede por casos:

- ✓ Si $(u, v) \in E_T$ entonces se tiene que (u, v) es seguro para A.
- ✓ Si $(u, v) \notin E_T$ entonces (u, v) forma un ciclo con las aristas del camino simple entre u y v en T. Dado que u y v están en lados opuestos del corte $(S, V \backslash S)$ al menos una arista de T hace parte del camino entre u y v y cruza el corte $(S, V \backslash S)$.

  Sea (x, y) dicha arista. (x, y) no está en A puesto que el corte $(S, V \backslash S)$ respeta A. Si se elimina (x, y) de T, este se divide en dos componentes. Si luego se añade (u, v) los dos componentes se reconectan y se obtiene un árbol de cubrimiento $T' = T - \{(x, y)\} \cup \{(u, v)\}$.

Árbol de Cubrimiento Mínimo

Generalidades

Demostración: Sea $T = (V, E_T)$ un MST tal que $A \subseteq E_T$. Se procede por casos:

- ✓ ...
- ✓ Para mostrar que T' es un MST se parte del hecho de que (u, v) es un eje ligero que cruza el corte $(S, V \backslash S)$ y (x, y) también lo cruza.

Árbol de Cubrimiento Mínimo

Generalidades

Demostración: Sea $T = (V, E_T)$ un MST tal que $A \subseteq E_T$. Se procede por casos:

- ✓ ...
- ✓ Para mostrar que T' es un MST se parte del hecho de que (u, v) es un eje ligero que cruza el corte $(S, V \backslash S)$ y (x, y) también lo cruza. Por lo tanto $w(u, v) \leq w(x, y)$. Luego:

$$\begin{aligned} w(T') = &\; w(T) - w(x, y) + w(u, v) \\ \leq &\; w(T) \end{aligned}$$

Árbol de Cubrimiento Mínimo

Generalidades

Demostración: Sea $T = (V, E_T)$ un MST tal que $A \subseteq E_T$. Se procede por casos:

- ✓ ...
- ✓ Para mostrar que T' es un MST se parte del hecho de que (u, v) es un eje ligero que cruza el corte $(S, V \backslash S)$ y (x, y) también lo cruza. Por lo tanto $w(u, v) \leq w(x, y)$. Luego:

$$\begin{aligned} w(T') = & \; w(T) - w(x, y) + w(u, v) \\ \leq & \; w(T) \end{aligned}$$

  Sin embargo, como T es un MST entonces $w(T) \leq w(T')$ y por consiguiente T' también es un MST. En consecuencia la arista (u, v) es segura puesto que hace parte de un MST.

Árbol de Cubrimiento Mínimo

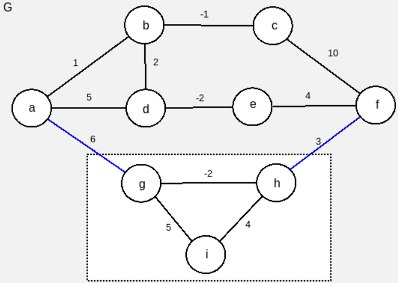
Propiedad de corte

Para cualquier corte del grafo G se cumple que la arista con menor peso en el conjunto de corte hace parte del árbol de cubrimiento mínimo de G.

Árbol de Cubrimiento Mínimo

Propiedad de corte

Para cualquier corte del grafo G se cumple que la arista con menor peso en el conjunto de corte hace parte del árbol de cubrimiento mínimo de G.

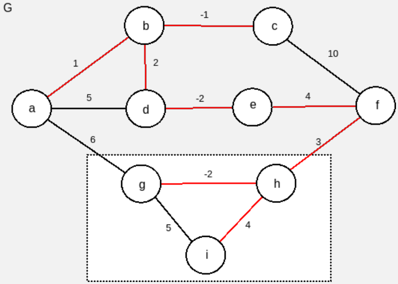


En el recuadro con líneas punteadas se denota un posible corte del grafo visto anteriormente.

Árbol de Cubrimiento Mínimo

Propiedad de corte

Para cualquier corte del grafo G se cumple que la arista con menor peso en el conjunto de corte hace parte del árbol de cubrimiento mínimo de G.



La arista f-h tiene menor peso y dicha arista está en el árbol de cubrimiento mínimo del grafo.

Plan

Árbol de Cubrimiento Mínimo

Algoritmo de Prim

- ✓ El algoritmo de Prim o también conocido como el algoritmo de Jarník es un algoritmo que permite encontrar un árbol de cubrimiento mínimo en un grafo.
- ✓ Fue propuesto por Vojtech Jarník en 1930. Aunque fue también redescubierto por Robert Prim en 1957 y Edsger Dijkstra en 1959.

Árbol de Cubrimiento Mínimo

Algoritmo de Prim

- ✓ El algoritmo de Prim o también conocido como el algoritmo de Jarník es un algoritmo que permite encontrar un árbol de cubrimiento mínimo en un grafo.
- ✓ Fue propuesto por Vojtech Jarník en 1930. Aunque fue también redescubierto por Robert Prim en 1957 y Edsger Dijkstra en 1959.
- ✓ Este algoritmo sigue una estrategia voraz y construye incrementalmente el árbol T agregando en cada iteración la arista (u, v) de menor peso tal que u pertenece a T y v no.

Árbol de Cubrimiento Mínimo

Algoritmo de Prim

- ✓ El algoritmo de Prim o también conocido como el algoritmo de Jarník es un algoritmo que permite encontrar un árbol de cubrimiento mínimo en un grafo.
- ✓ Fue propuesto por Vojtech Jarník en 1930. Aunque fue también redescubierto por Robert Prim en 1957 y Edsger Dijkstra en 1959.
- ✓ Este algoritmo sigue una estrategia voraz y construye incrementalmente el árbol T agregando en cada iteración la arista (u, v) de menor peso tal que u pertenece a T y v no.
- ✓ A partir de la propiedad de corte, se tiene que dicha arista debe pertenecer al árbol de cubrimiento mínimo.

Árbol de Cubrimiento Mínimo

Algoritmo de Prim

- ✓ El algoritmo de Prim o también conocido como el algoritmo de Jarník es un algoritmo que permite encontrar un árbol de cubrimiento mínimo en un grafo.
- ✓ Fue propuesto por Vojtech Jarník en 1930. Aunque fue también redescubierto por Robert Prim en 1957 y Edsger Dijkstra en 1959.
- ✓ Este algoritmo sigue una estrategia voraz y construye incrementalmente el árbol T agregando en cada iteración la arista (u, v) de menor peso tal que u pertenece a T y v no.
- ✓ A partir de la propiedad de corte, se tiene que dicha arista debe pertenecer al árbol de cubrimiento mínimo.
- ✓ Dependiendo de la estructuras de datos utilizadas, este algoritmo tiene complejidad $O(n^2)$, $O(m\ log\ n)$ o $O(m + n\ log\ n)$.

Árbol de Cubrimiento Mínimo

Algoritmo de Prim

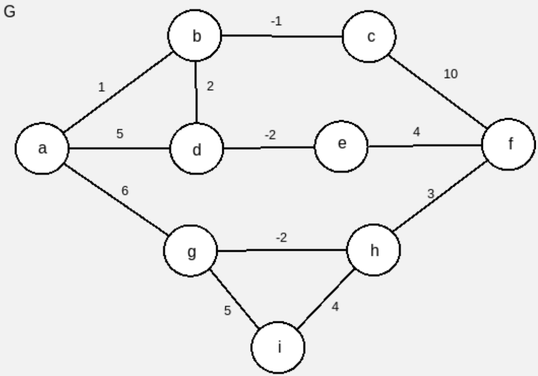
El siguiente es el código del algoritmo:

```
Prim(G, w):
Para cada vertice v en G.V:
  Hacer v.c = inf
  Hacer v.p = nil
Escoger un vertice r arbitrario y hacer r.c = 0
Agregar G.V a la cola Q
Mientras la cola Q no este vacia:
  Hacer u = extraer-minimo(Q)
  Para cada vertice v adyacente a u y que esta en la cola Q:
    Si w(u, v) < v.c:
      Hacer v.p = u y v.c = w(u, v)
```

Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo

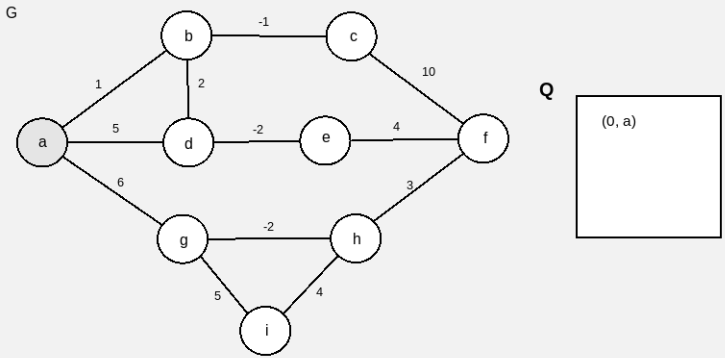
Considere el siguiente grafo:



Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo

Inicialmente, cada vértice está en un árbol diferente:



Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo

Inicialmente los valores de los atributos son:

Nodo	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
c	0	∞	∞	∞	∞	∞	∞	∞	∞

Árbol de Cubrimiento Mínimo

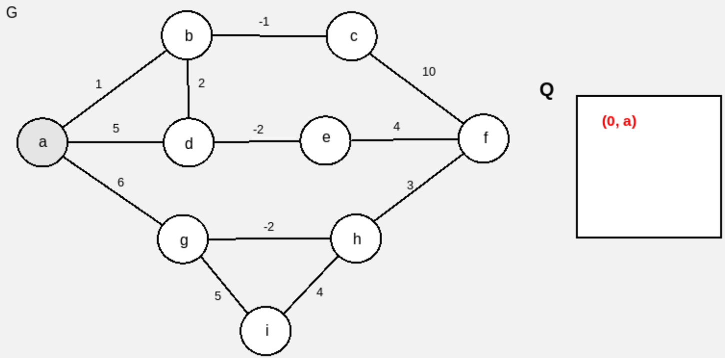
Algoritmo de Prim: Ejemplo

Inicialmente los predecesores son:

Nodo	a	b	c	d	e	f	g	h	i
p	-1	-1	-1	-1	-1	-1	-1	-1	-1

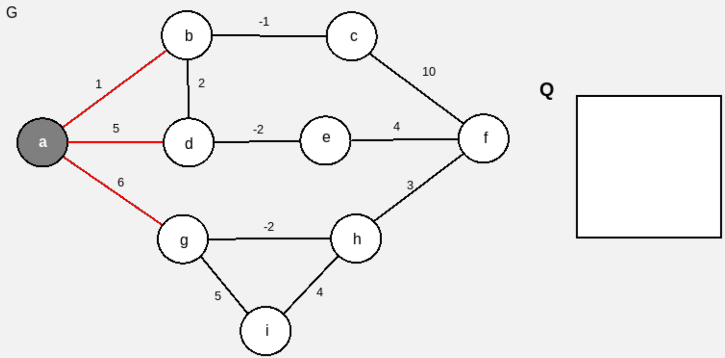
Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



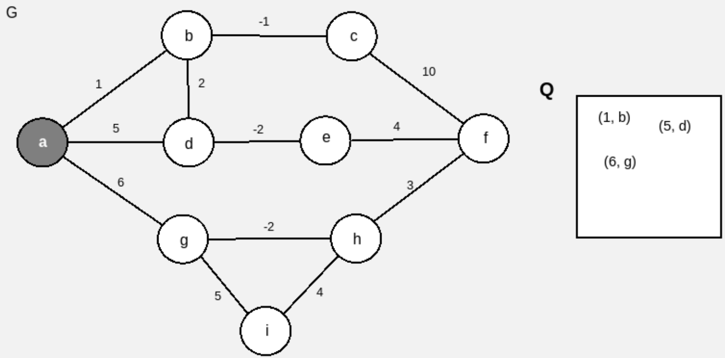
Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



Árbol de Cubrimiento Mínimo

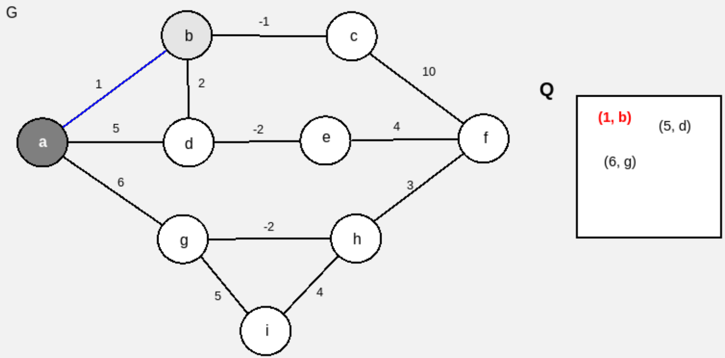
Algoritmo de Prim: Ejemplo

Ahora, los costos y los predecesores son:

Nodo	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
c	0	1	∞	5	∞	∞	6	∞	∞
pi	-1	a	-1	a	-1	-1	a	-1	-1

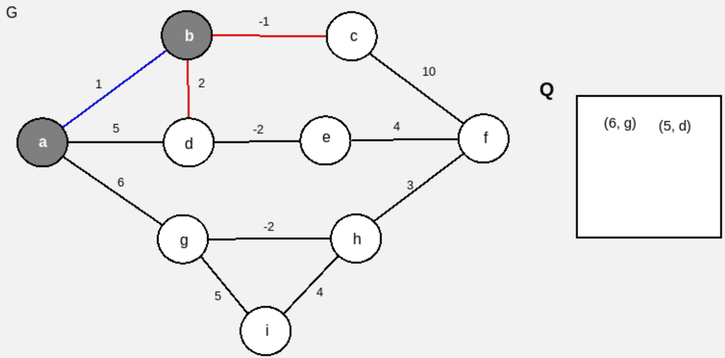
Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



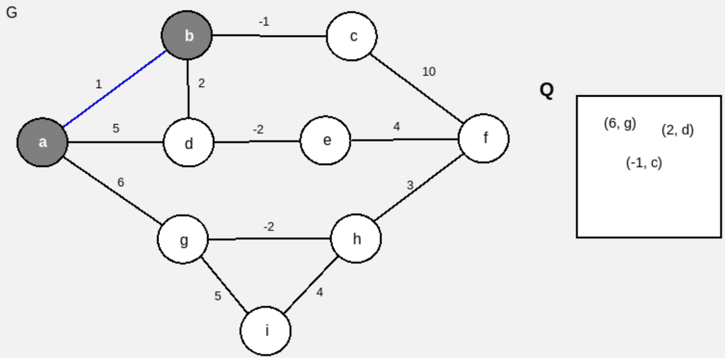
Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



Árbol de Cubrimiento Mínimo

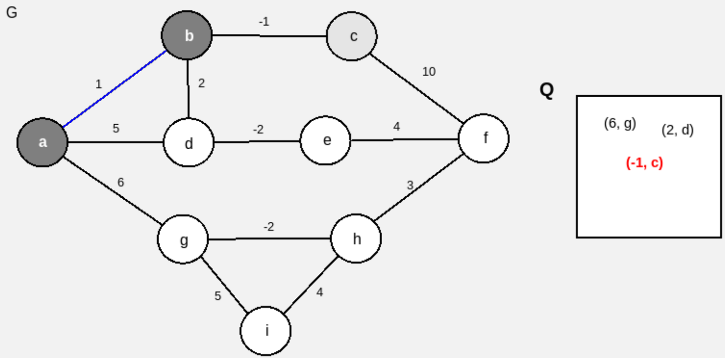
Algoritmo de Prim: Ejemplo

Ahora, los costos y los predecesores son:

Nodo	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
c	0	1	-1	2	∞	∞	6	∞	∞
pi	-1	a	b	b	-1	-1	a	-1	-1

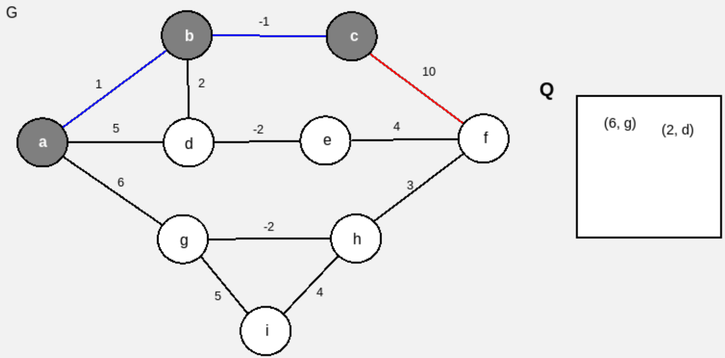
Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



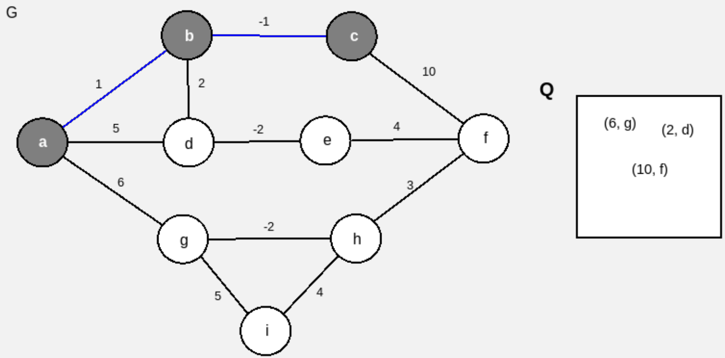
Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



Árbol de Cubrimiento Mínimo

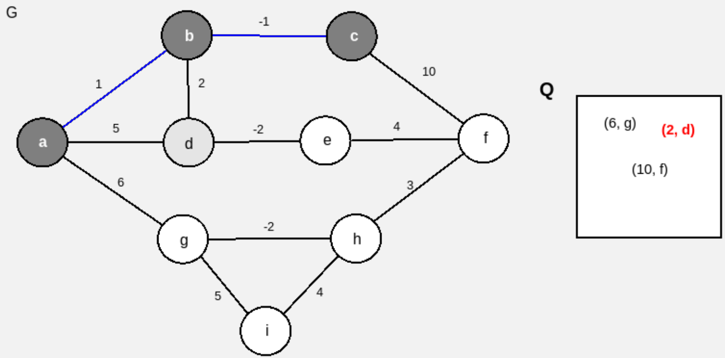
Algoritmo de Prim: Ejemplo

Ahora, los costos y los predecesores son:

Nodo	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
c	0	1	-1	2	∞	10	6	∞	∞
pi	-1	a	b	b	-1	c	a	-1	-1

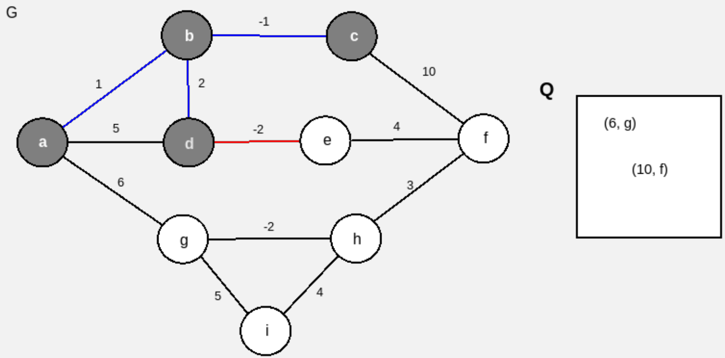
Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



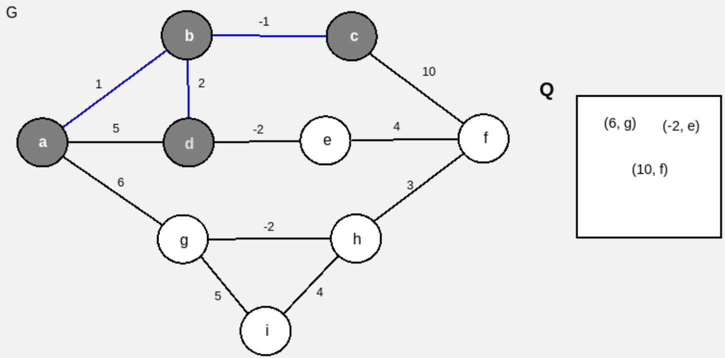
Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



Árbol de Cubrimiento Mínimo

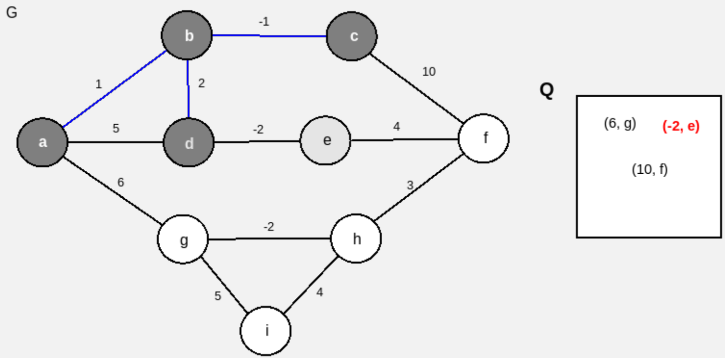
Algoritmo de Prim: Ejemplo

Ahora, los costos y los predecesores son:

Nodo	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
c	0	1	-1	2	-2	10	6	∞	∞
pi	-1	a	b	b	d	c	a	-1	-1

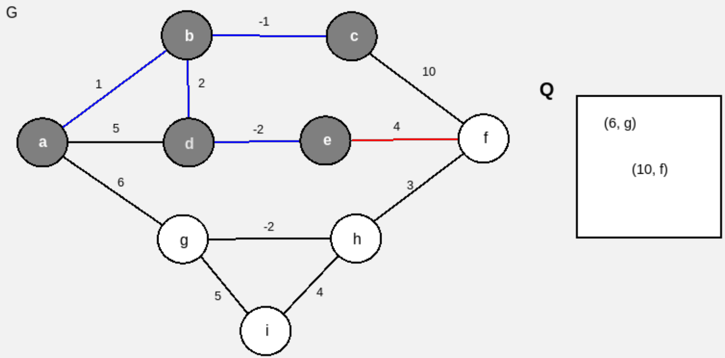
Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



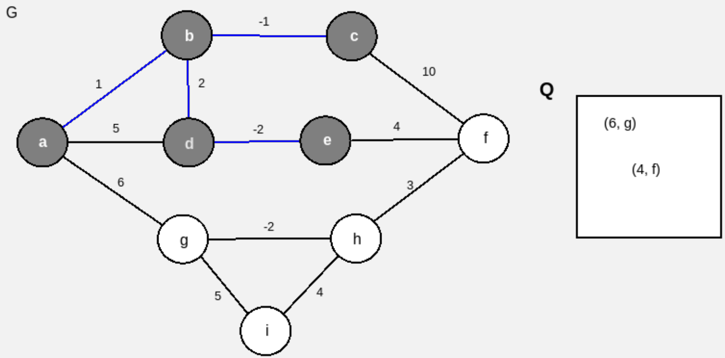
Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



Árbol de Cubrimiento Mínimo

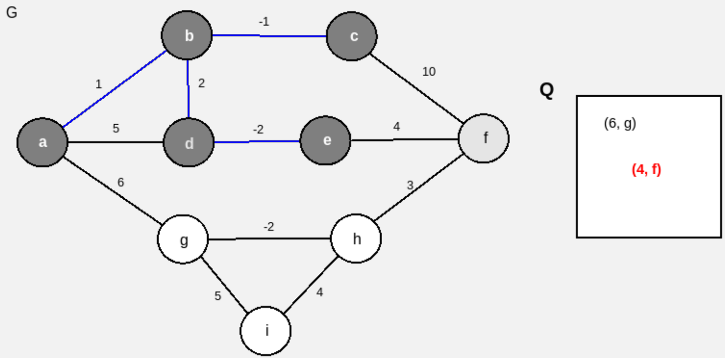
Algoritmo de Prim: Ejemplo

Ahora, los costos y los predecesores son:

Nodo	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
c	0	1	-1	2	-2	4	6	∞	∞
pi	-1	a	b	b	d	e	a	-1	-1

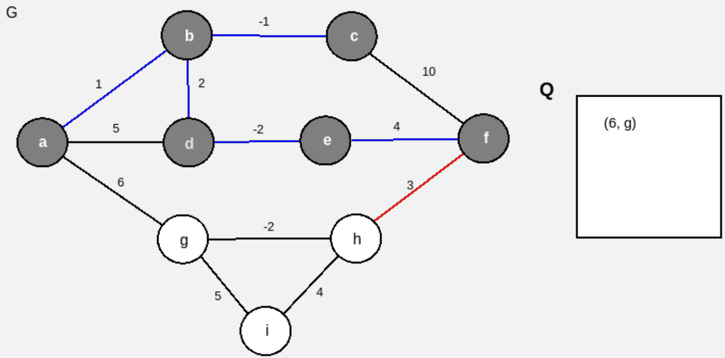
Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



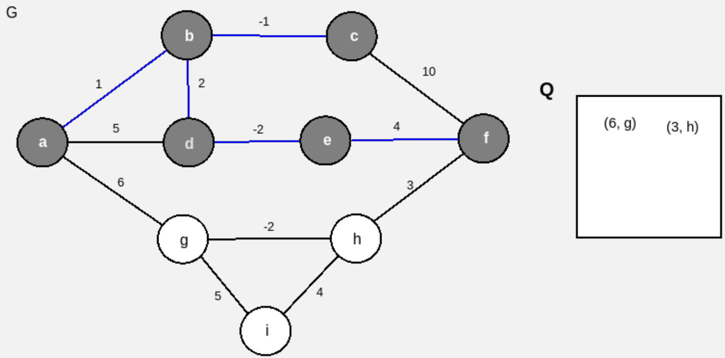
Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



Árbol de Cubrimiento Mínimo

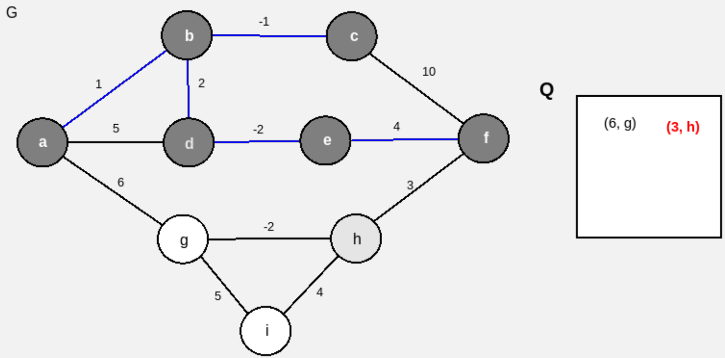
Algoritmo de Prim: Ejemplo

Ahora, los costos y los predecesores son:

Nodo	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
c	0	1	-1	2	-2	4	6	3	∞
pi	-1	a	b	b	d	e	a	f	-1

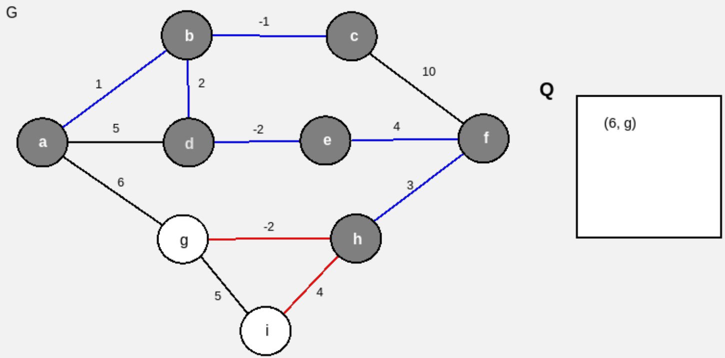
Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



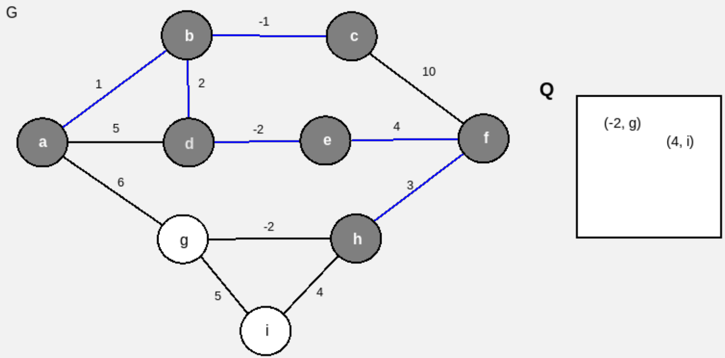
Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



Árbol de Cubrimiento Mínimo

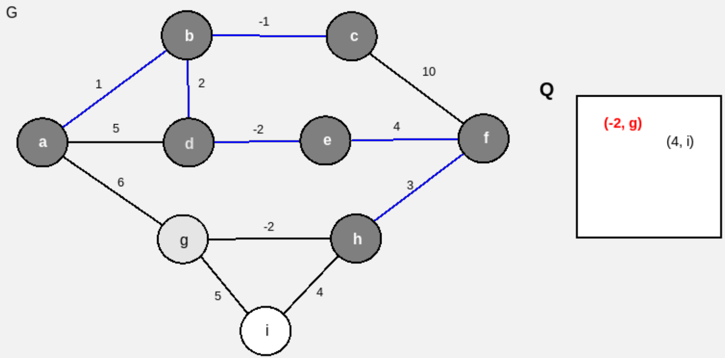
Algoritmo de Prim: Ejemplo

Ahora, los costos y los predecesores son:

Nodo	a	b	c	d	e	f	g	h	i
c	0	1	-1	2	-2	4	-2	3	4
pi	-1	a	b	b	d	e	h	f	h

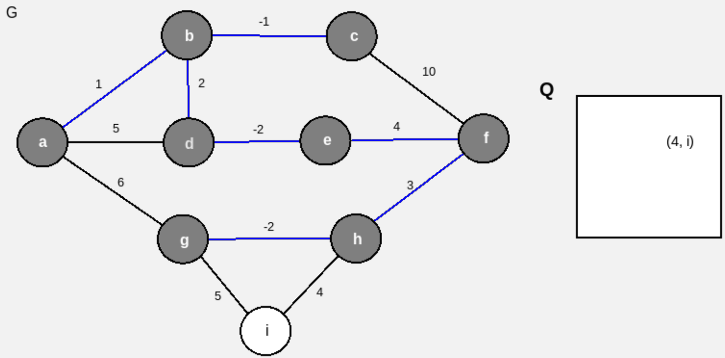
Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



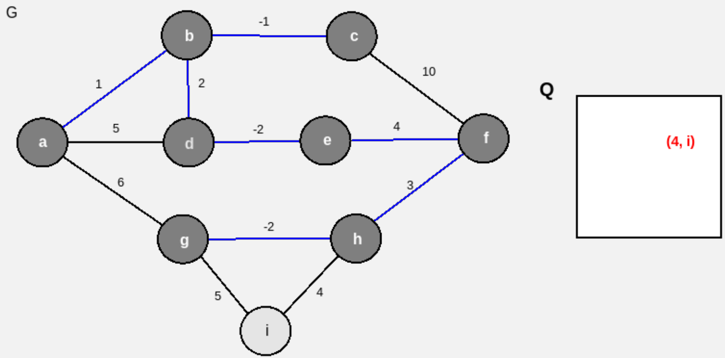
Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



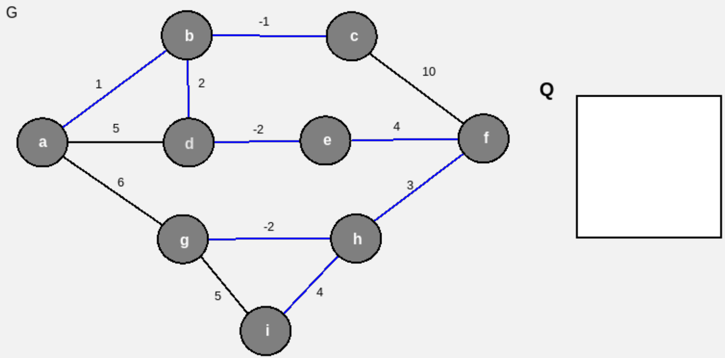
Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo



Árbol de Cubrimiento Mínimo

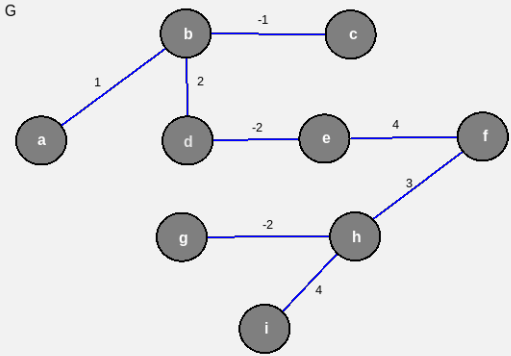
Algoritmo de Prim: Ejemplo



Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo

En consecuencia, el árbol de cubrimiento mínimo del grafo original es:



Árbol de Cubrimiento Mínimo

Algoritmo de Prim: Ejemplo

Finalmente, los costos y los predecesores son:

Nodo	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
c	0	1	-1	2	-2	4	-2	3	4
pi	-1	a	b	b	d	e	h	f	h

Plan

Plan

Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal

- ✓ El algoritmo de Kruskal es un algoritmo que permite encontrar un bosque de cubrimiento mínimo en un grafo.
- ✓ Fue propuesto por Joseph Kruskal en 1956.

Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal

- ✓ El algoritmo de Kruskal es un algoritmo que permite encontrar un bosque de cubrimiento mínimo en un grafo.
- ✓ Fue propuesto por Joseph Kruskal en 1956.
- ✓ Como el algoritmo de Prim, el algoritmo de Kruskal sigue una estrategia voraz pero permite calcular directamente el bosque cuando el grafo es no conexo.

Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal

- ✓ El algoritmo de Kruskal es un algoritmo que permite encontrar un bosque de cubrimiento mínimo en un grafo.
- ✓ Fue propuesto por Joseph Kruskal en 1956.
- ✓ Como el algoritmo de Prim, el algoritmo de Kruskal sigue una estrategia voraz pero permite calcular directamente el bosque cuando el grafo es no conexo.
- ✓ Este algoritmo ordena las aristas del grafo de acuerdo a su peso y va construyendo incrementalmente los diferentes árboles de cubrimiento mínimo que conforman el bosque.

Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal

- ✓ En cada paso usa la arista de menor peso que no ha sido usada: la agrega a un árbol o mediante ella conecta dos árboles.

Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal

- ✓ En cada paso usa la arista de menor peso que no ha sido usada: la agrega a un árbol o mediante ella conecta dos árboles.
- ✓ Inicialmente se asume que cada vértice en el grafo hace parte de un árbol de cubrimiento mínimo diferente.

Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal

- ✓ En cada paso usa la arista de menor peso que no ha sido usada: la agrega a un árbol o mediante ella conecta dos árboles.
- ✓ Inicialmente se asume que cada vértice en el grafo hace parte de un árbol de cubrimiento mínimo diferente.
- ✓ El algoritmo agrupa los vértices de cada árbol en conjuntos por lo que los vértices que pertenecen a diferentes árboles están en un conjunto diferente (conjuntos disyuntos).

Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal

- ✓ En cada paso usa la arista de menor peso que no ha sido usada: la agrega a un árbol o mediante ella conecta dos árboles.
- ✓ Inicialmente se asume que cada vértice en el grafo hace parte de un árbol de cubrimiento mínimo diferente.
- ✓ El algoritmo agrupa los vértices de cada árbol en conjuntos por lo que los vértices que pertenecen a diferentes árboles están en un conjunto diferente (conjuntos disyuntos).
- ✓ Si se utilizan conjuntos disyuntos con una implementación eficiente (*union-find disjoint set*), este algoritmo tiene complejidad $O(m\ log\ n)$.

Plan

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos

- ✓ Como se mencionó anteriormente, la implementación del algoritmo de Kruskal depende de la implementación eficiente de conjuntos disyuntos.

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos

- ✓ Como se mencionó anteriormente, la implementación del algoritmo de Kruskal depende de la implementación eficiente de conjuntos disyuntos.
- ✓ Una estructura de datos de conjuntos disyuntos mantiene una colección de conjuntos $S = \{S_1, S_2, \ldots, S_k\}$ de conjuntos disyuntos dinámicos.

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos

- ✓ Como se mencionó anteriormente, la implementación del algoritmo de Kruskal depende de la implementación eficiente de conjuntos disyuntos.
- ✓ Una estructura de datos de conjuntos disyuntos mantiene una colección de conjuntos $S = \{S_1, S_2, \ldots, S_k\}$ de conjuntos disyuntos dinámicos.
- ✓ Cada conjunto se identifica mediante un elemento representativo.

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos

- ✓ En algunas aplicaciones, no importa cúal elemento de cada conjunto es usado como elemento representativo.
- ✓ Solo es necesario que al consultar dos veces sin que hayan modificaciones entre ellas, se obtenga la misma respuesta en ambos casos.
- ✓ Sin embargo, en otras aplicaciones se requiere una regla para escoger el elemento representativo, por ejemplo, escoger el elemento más pequeño.

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos

Sea x un elemento, la estructura de datos de conjuntos disyuntos soporta las siguientes operaciones:

- ✓ `Make-Set`(x): Esta operación crea un nuevo conjunto cuyo único elemento (y por consiguiente representativo) es x. Dado que los conjuntos son disyuntos, es requerido que x no esté en los otros conjuntos.

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos

Sea x un elemento, la estructura de datos de conjuntos disyuntos soporta las siguientes operaciones:

- ✓ `Make-Set`(x): Esta operación crea un nuevo conjunto cuyo único elemento (y por consiguiente representativo) es x. Dado que los conjuntos son disyuntos, es requerido que x no esté en los otros conjuntos.
- ✓ `Find-Set`(x): Esta operación retorna un puntero al elemento representativo del conjunto que contiene a x.

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos

Sea x un elemento, la estructura de datos de conjuntos disyuntos soporta las siguientes operaciones:

- ✓ `Make-Set`(x): Esta operación crea un nuevo conjunto cuyo único elemento (y por consiguiente representativo) es x. Dado que los conjuntos son disyuntos, es requerido que x no esté en los otros conjuntos.
- ✓ `Find-Set`(x): Esta operación retorna un puntero al elemento representativo del conjunto que contiene a x.
- ✓ `Union`(x, y): Esta operación une los conjuntos que contienen a x y y, denotados como S_x y S_y. El elemento representativo del conjunto resultante es cualquier elemento.

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos

- ✓ Existen diversas alternativas de implementación para la estructura de conjuntos disyuntos incluyendo arreglos y listas doblemente enlazadas.

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos

- ✓ Existen diversas alternativas de implementación para la estructura de conjuntos disyuntos incluyendo arreglos y listas doblemente enlazadas.
- ✓ Sin embargo, existe una estrategia de implementación que se ha probado es asintóticamente óptima.

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos

- ✓ Existen diversas alternativas de implementación para la estructura de conjuntos disyuntos incluyendo arreglos y listas doblemente enlazadas.
- ✓ Sin embargo, existe una estrategia de implementación que se ha probado es asintóticamente óptima.
- ✓ La estrategia de bosques de conjuntos disyuntos junto a las heurísticas de compresión de camino y de unión por rango tiene una complejidad $O(m \cdot \alpha(n))$ que es cercana a ser lineal.

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos

- ✓ Existen diversas alternativas de implementación para la estructura de conjuntos disyuntos incluyendo arreglos y listas doblemente enlazadas.
- ✓ Sin embargo, existe una estrategia de implementación que se ha probado es asintóticamente óptima.
- ✓ La estrategia de bosques de conjuntos disyuntos junto a las heurísticas de compresión de camino y de unión por rango tiene una complejidad $O(m \cdot \alpha(n))$ que es cercana a ser lineal.
- ✓ α es la función inversa de Ackermann y m es el número total de operaciones que se realiza sobre la estructura.

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos

- ✓ Para cada uno de los elementos x se registra un atributo `p` y un atributo `rango`.

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos

- ✓ Para cada uno de los elementos x se registra un atributo `p` y un atributo `rango`.
- ✓ El atributo `p` corresponde a un identificador del padre de x en el árbol al cual pertenece (elemento representativo del conjunto).
- ✓ Sin embargo, con la heurística de compresión de camino, el atributo `p` apunta directamente a la raíz del árbol (operación `Find-Set`).

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos

- ✓ Para cada uno de los elementos x se registra un atributo `p` y un atributo `rango`.
- ✓ El atributo `p` corresponde a un identificador del padre de x en el árbol al cual pertenece (elemento representativo del conjunto).
- ✓ Sin embargo, con la heurística de compresión de camino, el atributo `p` apunta directamente a la raíz del árbol (operación `Find-Set`).
- ✓ Con la heurística de unión por rango, cuando se unen dos árboles,la raíz del elemento con menor rango apuntará a la raíz del elemento con mayor rango (operación `Union`).

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos

```
Make-Set(x):
1. Hacer x.p = x
2. Hacer x.rango = 0

Union(x, y):
1. Si x.rango > y.rango:
2.   Hacer y.p = x
3. De lo contrario x.p = y
4.   Si x.rango == y.rango:
5.     y.rango == y.rango + 1

Find-Set(x):
1. if x != x.p:
2.   Hacer x.p = Find-Set(x.p)
3. Retornar x.p
```

Árbol de Cubrimiento Mínimo

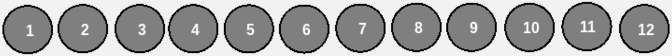
Estructura de Conjuntos Disyuntos: Ejemplo

- ✓ Considere que se tienen los números del 1 al 12 como elementos de la estructura de conjuntos disyuntos.

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos: Ejemplo

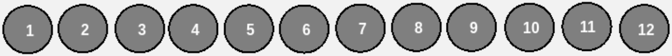
- ✓ Considere que se tienen los números del 1 al 12 como elementos de la estructura de conjuntos disyuntos.
- ✓ Inicialmente, se efectua la operación `Make-Set(i)` para cada `i` entre 1 y 12, luego:



Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos: Ejemplo

- ✓ Considere que se tienen los números del 1 al 12 como elementos de la estructura de conjuntos disyuntos.
- ✓ Inicialmente, se efectua la operación `Make-Set(i)` para cada `i` entre 1 y 12, luego:



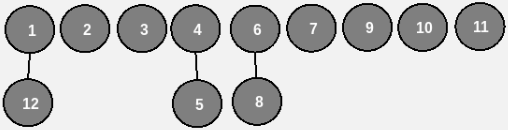
- ✓ Además, se tienen los siguientes datos:

Valor	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
p	1	2	3	4	5	6	7	8	9	10	11	12
rango	0	0	0	0	0	0	0	0	0	0	0	0

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos: Ejemplo

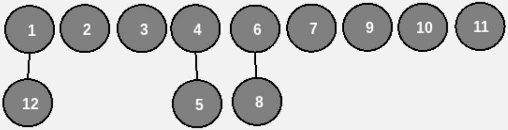
- ✓ Después de realizar las operaciones `Union(4,5)`, `Union(6,8)` y `Union(1, 12)` se tienen los siguientes árboles:



Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos: Ejemplo

- ✓ Después de realizar las operaciones `Union(4,5)`, `Union(6,8)` y `Union(1, 12)` se tienen los siguientes árboles:



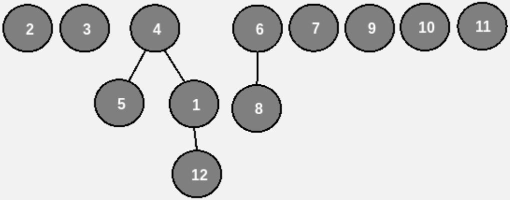
- ✓ Estos árboles son representados con los siguientes datos:

Valor	1	2	3	4	5	6	7	8	9	10	11	12
p	1	2	3	4	4	6	7	6	9	10	11	1
rango	1	0	0	1	0	1	0	0	0	0	0	0

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos: Ejemplo

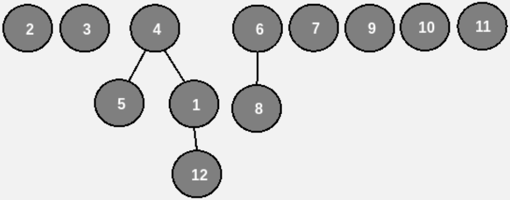
- ✓ Después de realizar las operación `Union(5,1)` se tienen los siguientes árboles:



Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos: Ejemplo

- ✓ Después de realizar las operación `Union(5,1)` se tienen los siguientes árboles:



- ✓ Estos árboles son representados con los siguientes datos:

Valor	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
p	4	2	3	4	4	6	7	6	9	10	11	1
rango	1	0	0	2	0	1	0	0	0	0	0	0

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos: Ejemplo

- ✓ Cuando se realiza la operación `Find-Set(12)` se obtiene 4.

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos: Ejemplo

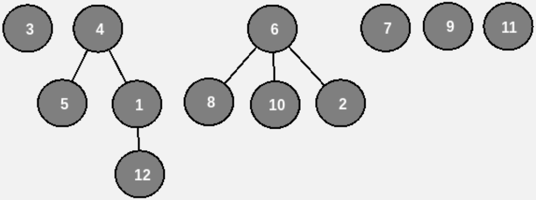
- ✓ Cuando se realiza la operación `Find-Set(12)` se obtiene 4.
- ✓ Sin embargo, al realizar esta operación se hace la compresión del camino de 12 y ahora se tienen los siguientes datos:

Valor	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
p	4	2	3	4	4	6	7	6	9	10	11	4
rango	1	0	0	2	0	1	0	0	0	0	0	0

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos: Ejemplo

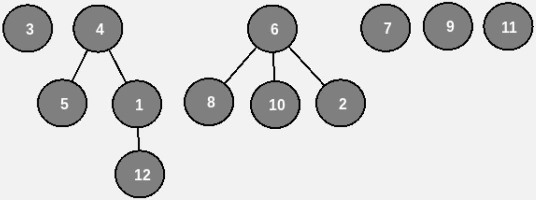
- ✓ Después de realizar las operaciones `Union(10,8)` y `Union(2, 6)` se tienen los siguientes árboles:



Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos: Ejemplo

- ✓ Después de realizar las operaciones `Union(10,8)` y `Union(2, 6)` se tienen los siguientes árboles:



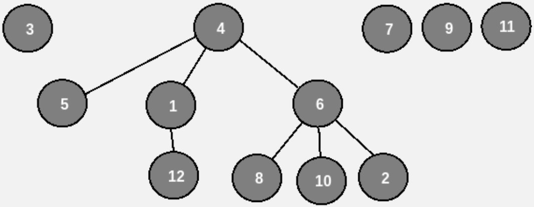
- ✓ Estos árboles son representados con los siguientes datos:

Valor	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
p	4	6	3	4	4	6	7	6	9	6	11	4
rango	1	0	0	2	0	1	0	0	0	0	0	0

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos: Ejemplo

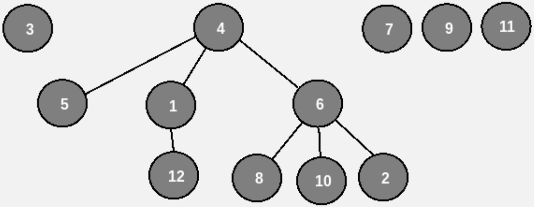
- ✓ Después de realizar las operaciones `Union(10,1)` se tienen los siguientes árboles:



Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos: Ejemplo

- Después de realizar las operaciones `Union(10,1)` se tienen los siguientes árboles:



- Estos árboles son representados con los siguientes datos:

Valor	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
p	4	6	3	4	4	4	7	6	9	6	11	4
rango	1	0	0	2	0	1	0	0	0	0	0	0

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos: Ejemplo

✓ Cuando se realiza la operación `Find-Set(8)` se obtiene 4.

Árbol de Cubrimiento Mínimo

Estructura de Conjuntos Disyuntos: Ejemplo

- ✓ Cuando se realiza la operación `Find-Set(8)` se obtiene 4.
- ✓ Sin embargo, al realizar esta operación se hace la compresión del camino de 8 y ahora se tienen los siguientes datos:

Valor	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
p	4	6	3	4	4	4	7	4	9	6	11	4
rango	1	0	0	2	0	1	0	0	0	0	0	0

Plan

Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal

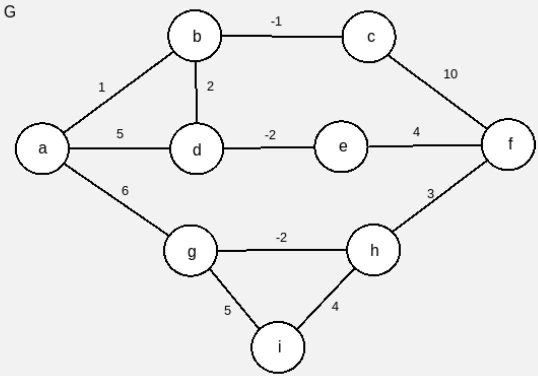
El siguiente es el código del algoritmo:

```
Kruskal(G, w):
1.  Hacer A = empty
2.  Para cada vertice v en G.V:
3.    Make-Set(v)
4.  Ordenar G.E ascendentemente con respecto al peso
6.  Para cada arista (u, v) en G.E:
7.    Hacer a1 = Find-Set(u) y a2 = Find-Set(v)
8.    Si a1 != a2:
9.      Hacer A = A union {(u, v)}
10.     Union(a1, a2)
11. Retornar A
```

Árbol de Cubrimiento Mínimo

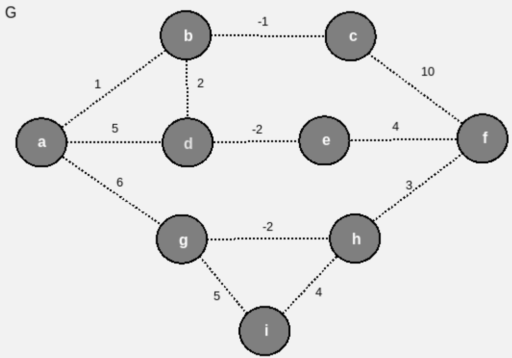
Algoritmo de Kruskal: Ejemplo

Considere el siguiente grafo:



Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo



Árbol de Cubrimiento Mínimo

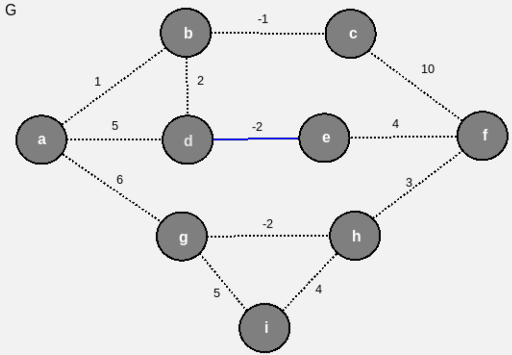
Algoritmo de Kruskal: Ejemplo

Inicialmente se tiene para cada nodo la siguiente información:

Nodo	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
conj.	0	1	2	3	4	5	6	7	8

Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo



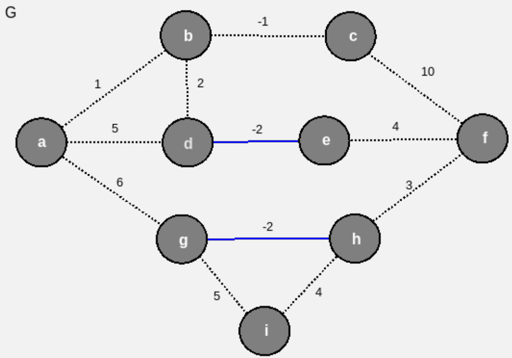
Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo

Nodo	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
conj.	0	1	2	3	3	5	6	7	8

Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo



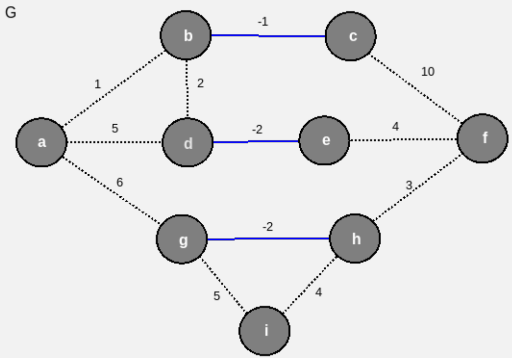
Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo

Nodo	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
conj.	0	1	2	3	3	5	6	6	8

Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo



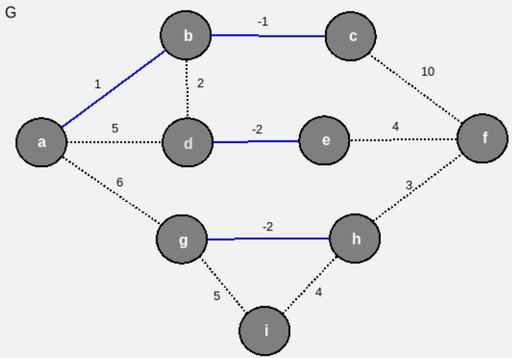
Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo

Nodo	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
conj.	0	1	1	3	3	5	6	6	8

Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo



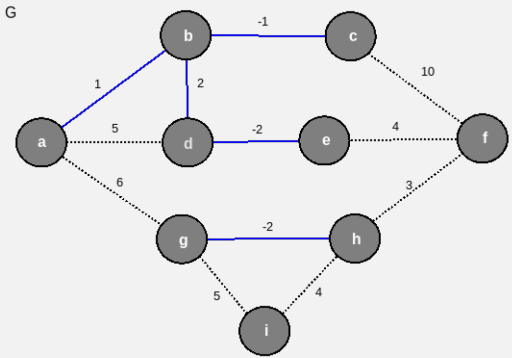
Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo

Nodo	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
conj.	1	1	1	3	3	5	6	6	8

Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo



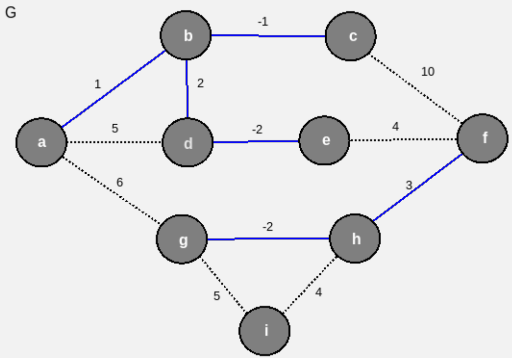
Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo

Nodo	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
conj.	1	1	1	1	1	5	6	6	8

Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo



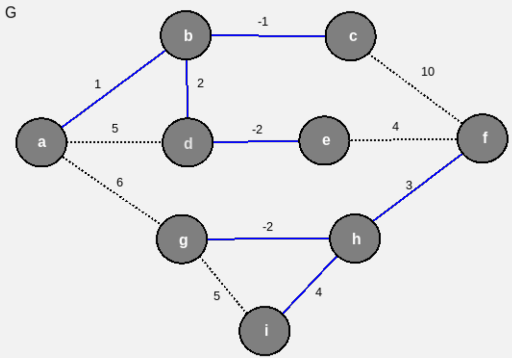
Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo

Nodo	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
conj.	1	1	1	1	1	6	6	6	8

Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo



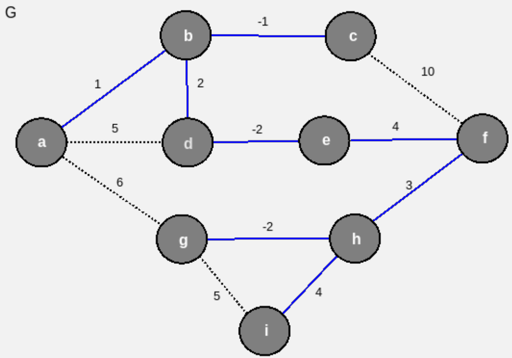
Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo

Nodo	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
conj.	1	1	1	1	1	6	6	6	6

Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo



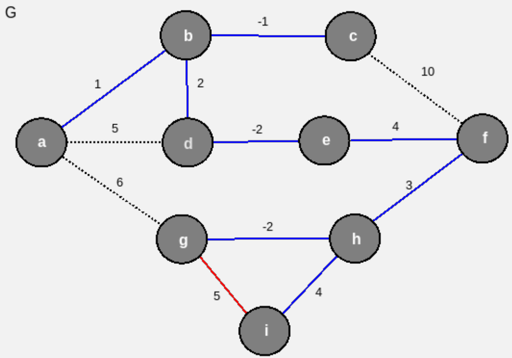
Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo

Nodo	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
conj.	1	1	1	1	1	1	1	1	1

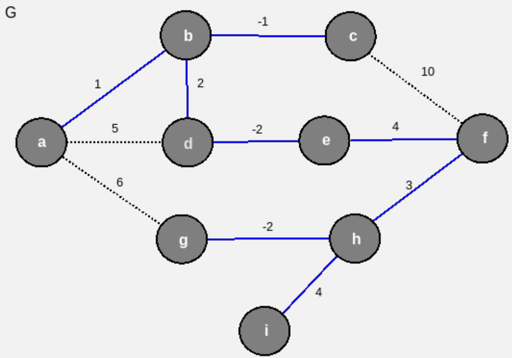
Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo



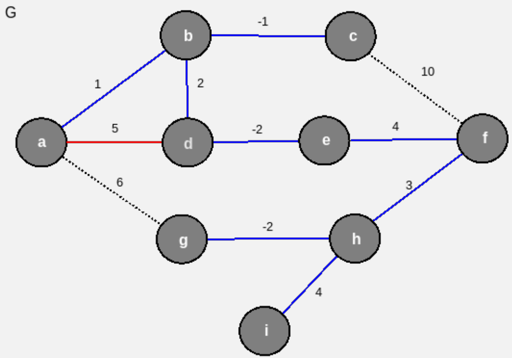
Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo



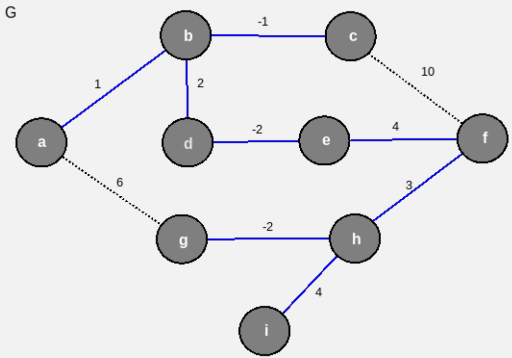
Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo



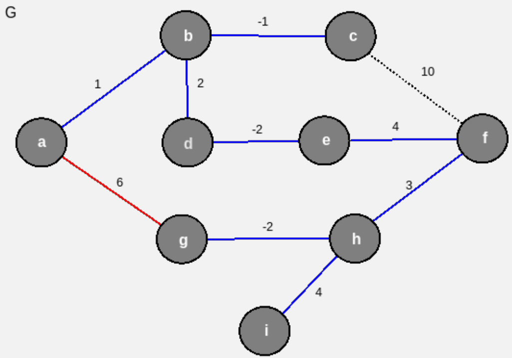
Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo



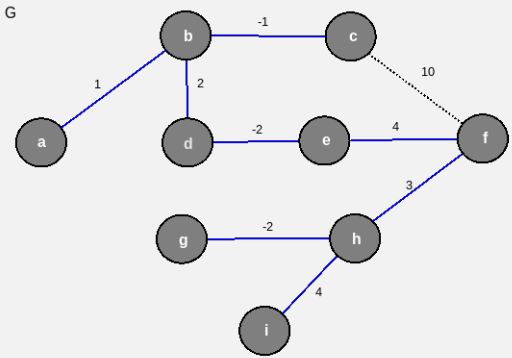
Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo



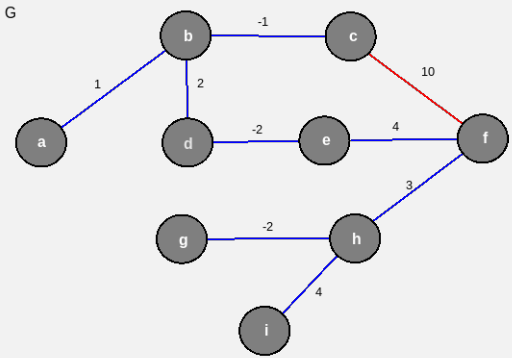
Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo



Árbol de Cubrimiento Mínimo

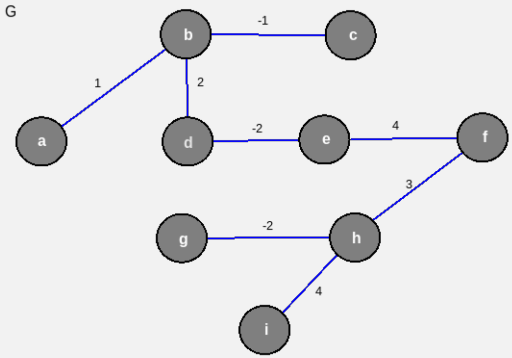
Algoritmo de Kruskal: Ejemplo



Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo

En consecuencia, el árbol de cubrimiento mínimo del grafo original es:



Árbol de Cubrimiento Mínimo

Algoritmo de Kruskal: Ejemplo

Finalmente, los datos son los siguientes:

Nodo	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
conj.	1	1	1	1	1	1	1	1	1

Preguntas

?