KRUSKAL (V 3.0)

Prof. José Fager

Montevideo julio de 2019

Joseph Kruskal

Imagen tomada de:
http://optimizacionenteraydinamicabibliofh.blogspot.com.uy/

"(29 de enero de 1928, 19 de septiembre de 2010) fue un matemático y estadístico estadounidense…"

Tomado de https://es.wikipedia.org/wiki/Joseph_Kruskal

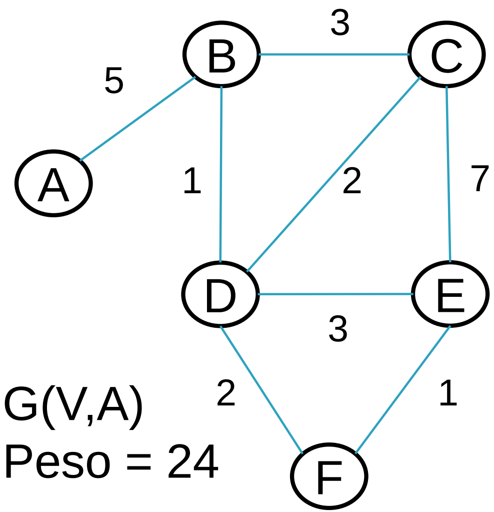
Si bien es común pronunciar "crúscal" tal cual se lee en castellano, en inglés se pronuncia: "cráscal" (https://es.forvo.com/word/joseph_kruskal/).

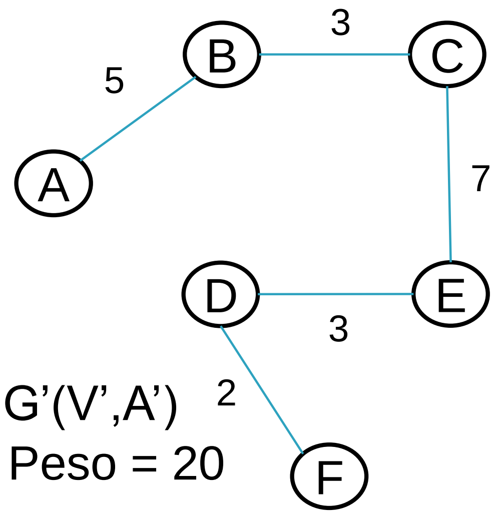
Definiciones

Árbol recubridor (AR)

- Dado un grafo conexo G, conformado por el conjunto de vértices V y el conjunto de aristas A:
  - G(V,A),
  - un AR de G es un grafo conexo G´ sin ciclos simples, conformado por el conjunto de vértices V’ y aristas A’:
  - G’(V’,A’),
  - G’ cumple que: V’ es igual a V y A’ está contenido en A.

Ejemplo de AR



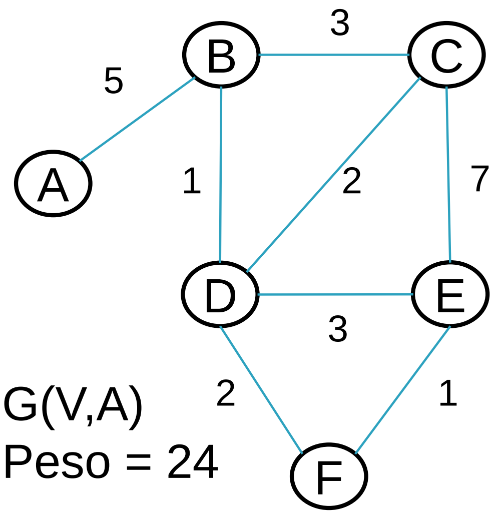


G' es un ejemplo de AR de G, G' es conexo, tienen los mismos vértices que G y las aristas de G' son un sub-conjunto de las aristas de G. G' no tiene ciclos simples.

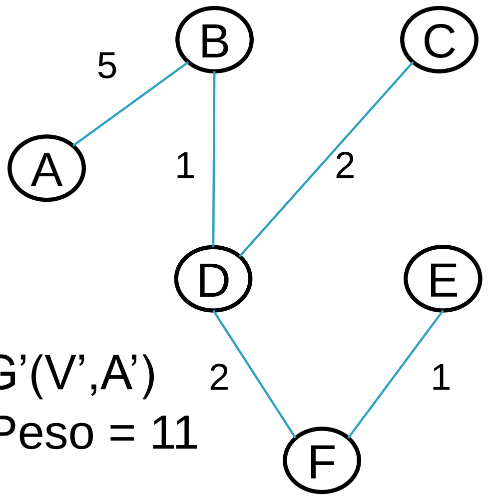
Árbol recubridor de peso mínimo (ARPM)

- Dado un grafo conexo G conformado por el conjunto de vértices V y aristas A, su ARPM es el AR de G de menor peso.

Ejemplo de (ARPM)



G(V,A)
Peso = 24



G'(V',A')
Peso = 11

G' es un ejemplo de ARPM de G, es un AR, es conexo, tienen los mismos vértices que G y sus aristas son un sub-conjunto de las aristas de G.

Pero además es el AR de G de menor peso que se puede encontrar.

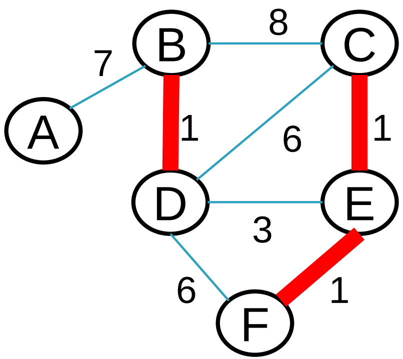
Algoritmo para encontrar el ARPM

Algoritmo de Kruskal

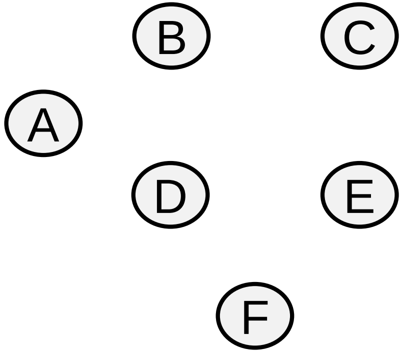
- El algoritmo de Kruskal permite obtener dado un grafo conexo G(V,A), el grafo conexo G’(V’, A’), que es el ARPM de G.
- **Paso base**
  - G’(V’,A’)   con V’=V y A’ está vacío (no tiene aristas).
- **Paso iterativo**
  - Mientras G’ no sea conexo hacer:
    - Tomar la arista x de menor peso de A ,que aún no fue considerada.
    - Agregar x a A’ si no forma un ciclo simple en G’.
- **Paso final**
  - G’(V’,A’) es conexo, entonces es el ARPM de G(V,A).

Ejemplo del algoritmo de Kruskal

Kruskal – paso I (paso base)



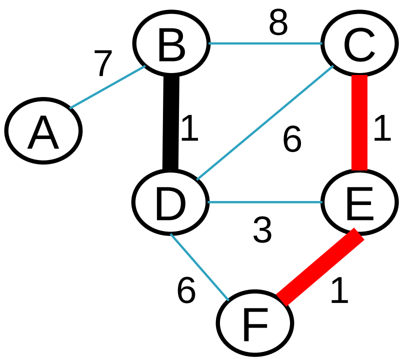
G(V,A)
V = {A, B, C, D, E, F}
A = {(A, B); (B, C); (B, D); (C, E); (C, D); (D, E); (D, F); (E, F)}



G’(V’,A’)
V’ = {A, B, C, D, E, F}
A’ ={}

V’ = V y no varía en toda la ejecución del algoritmo.
El conjunto A contiene a todas la aristas del grafo y tampoco varía.
El conjunto A’ inicialmente está vacío.

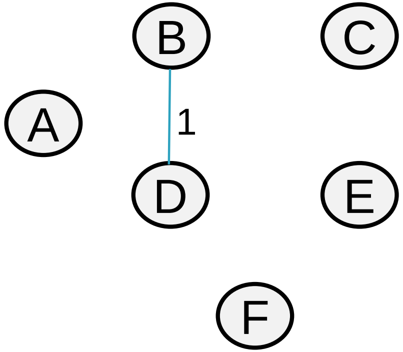
Kruskal – paso II (iterativo)



G(V,A)

V = {A, B, C, D, E, F}

A = {(A, B); (B, C); (B, D); (C, E); (C, D); (D, E); (D, F); (E, F)}



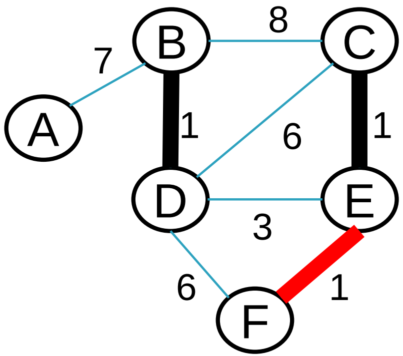
G'(V',A')

V' = {A, B, C, D, E, F}

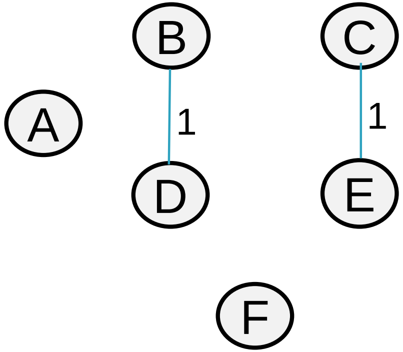
A' ={(B, D)}

Las posibles aristas a agregar son (B, D), (C, E) y (E, F), cualquiera se puede agregar pues no generan ciclos simples, se agrega (B, D).
Como G' sigue sin ser conexo se continúa con la ejecución del algoritmo.

Kruskal – paso III (iterativo)



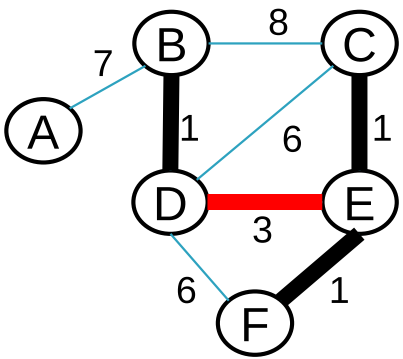
G(V,A)
V = {A, B, C, D, E, F}
A = {(A, B); (B, C); (B, D); (C, E); (C, D); (D, E); (D, F); (E, F)}



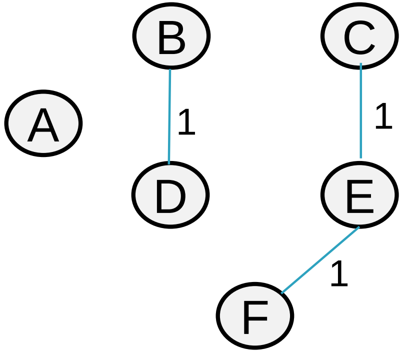
G'(V',A')
V' = {A, B, C, D, E, F}
A' ={(B, D); (C, E)}

Las posibles aristas a agregar son (C, E) y (E, F), cualquiera se puede agregar pues no generan ciclos simples, se agrega (C, E).
Como G' sigue sin ser conexo se continúa con la ejecución del algoritmo.

Kruskal – paso IV (iterativo)



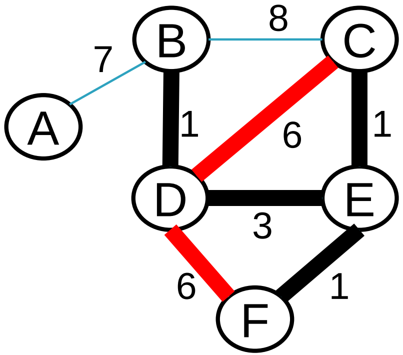
<u>G(V,A)</u>
V = {A, B, C, D, E, F}
A = {(A, B); (B, C); (B, D); (C, E); (C, D); (D, E); (D, F); (E, F)}



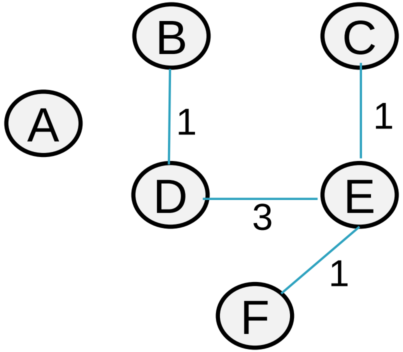
<u>G’(V’,A’)</u>
V’ = {A, B, C, D, E, F}
A’ ={(B, D); (C, E); (E, F)}

La única posible arista a agregar es (E, F), se puede agregar pues no genera ciclos simples.
Como G' sigue sin ser conexo se continúa con la ejecución del algoritmo.

Kruskal – paso V (iterativo)



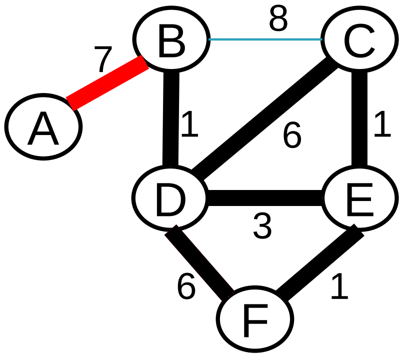
G(V,A)
V = {A, B, C, D, E, F}
A = {(A, B); (B, C); (B, D); (C, E); (C, D); (D, E); (D, F); (E, F)}



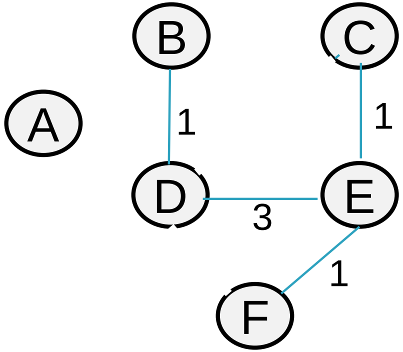
G'(V',A')
V' = {A, B, C, D, E, F}
A' ={(B, D); (C, E); (E, F); (D, E)}

La única posible arista a agregar es (D, E), se puede agregar pues no genera ciclos simples.
Como G' sigue sin ser conexo se continúa con la ejecución del algoritmo.

Kruskal – paso VI (iterativo)



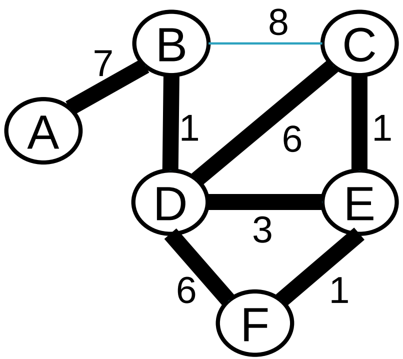
G(V,A)
V = {A, B, C, D, E, F}
A = {(A, B); (B, C); (B, D); (C, E); (C, D); (D, E); (D, F); (E, F)}



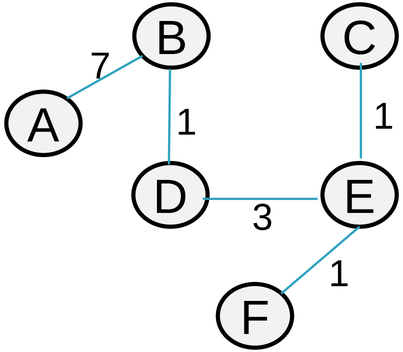
G'(V',A')
V' = {A, B, C, D, E, F}
A' ={(B, D); (C, E); (E, F); (D, E)}

Las posibles arista a agregar son (C, D) y (D, F), pero cualquiera de las dos generan un ciclo simple por lo tanto no podemos agregarlas.
Como el grafo sigue sin ser conexo se continúa la ejecución del algoritmo.

Kruskal – paso VII (iterativo)



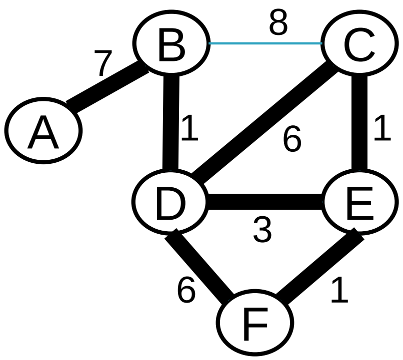
G(V,A)
V = {A, B, C, D, E, F}
A = {(A, B); (B, C); (B, D); (C, E); (C, D); (D, E); (D, F); (E, F)}



G'(V',A')
V' = {A, B, C, D, E, F}
A' ={(B, D); (C, E); (E, F); (D, E); (A, B)}

Ahora seleccionamos la arista (A, B) y como no genera un ciclo simple podemos agregarla. Todavía hay aristas sin considerar, pero como el grafo ahora es conexo se termina la ejecución del algoritmo.

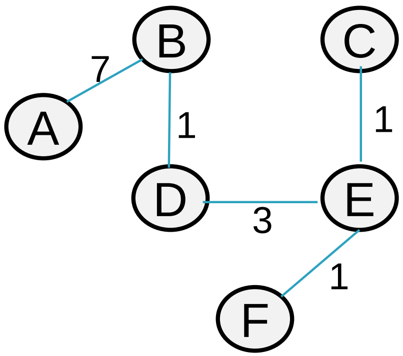
Kruskal – paso VIII (final)



G(V,A)

V = {A, B, C, D, E, F}

A = {(A, B); (B, C); (B, D); (C, E); (C, D); (D, E); (D, F); (E, F)}



G’(V’,A’)

V’ = {A, B, C, D, E, F}

A’ ={(B, D); (C, E); (E, F); (D, E); (A, B)}

El ARPM de G es G’ y su peso es: 7+1+3+1+1= 13.