Algoritmo de KRUSKAL

Algoritmo de Kruskal (Conceptos)

- Este algoritmo fue publicado en los *Proceedings of the American Mathematical Society*, pp.48–50 en 1956,por Joseph Kruskal.

- Es un algoritmo de la teoría de grafos para encontrar un árbol recubridor mínimo, también llamado árbol de expansión, en un grafo conexo, no dirigido y ponderado.

- **El objetivo del algoritmo de Kruskal** es construir un árbol (subgrafo sin ciclos) formado por aristas sucesivamente seleccionados de mínimo peso a partir de un grafo ponderado.

Algoritmo de Kruskal (Conceptos)

- Un árbol (spanning tree) de un grafo es un subgrafo que contiene todos sus vértices o nodos. Un grafo puede tener múltiples árboles. Por ejemplo, un grafo completo de cuatro nodos (todos relacionados con todos) tendría 16 árboles.

- Este algoritmo busca las distancias más cortas (de menor costo) de un árbol expandido que conectan todos los vértices. Siendo así, una variación del algoritmo de mínima conexión.

Algoritmo de Kruskal (Conceptos)

- La idea principal de kruskal consiste en ser ambicioso, escogiendo siempre la arista menos costosa disponible y cuidando que en cada paso del proceso no se forme ningún circuito.
- La aplicación típica de este problema es el diseño de redes telefónicas. Una empresa con diferentes oficinas, trata de trazar líneas de teléfono para conectarlas unas con otras. La compañía telefónica le ofrece está interconexión, pero ofrece tarifas diferentes o costos por conectar cada par de oficinas. ¿Cómo conectar entonces las oficinas al mínimo costo total?

Procedimiento del algoritmo de Kruskal

1. Ordenar en forma ascendente según el peso de cada arista.

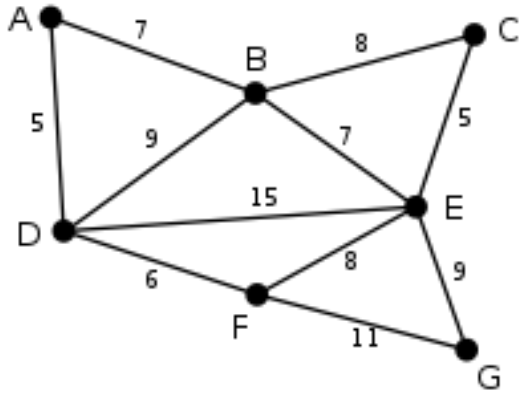
2. Se marca la arista con menor peso como la arista inicial de la ruta mínima.

3. De las aristas restantes, se selecciona la que tenga menor peso, si hay más de una, se elige cualquiera de ellas.

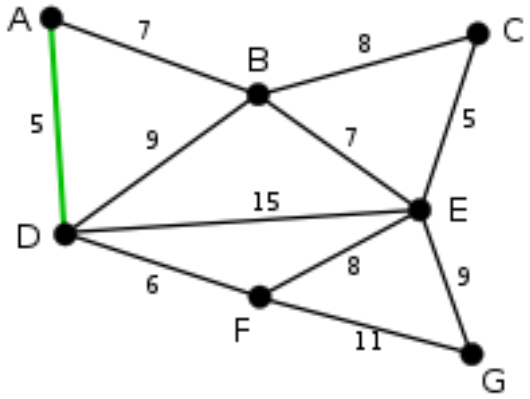
4. Repetir el paso3 siempre que la arista elegida no forme un ciclo con las ya seleccionadas.

5. El proceso termina cuando tenemos todos los nodos del grafo en alguna de las aristas marcadas, es decir, cuando tenemos seleccionadas n-1 aristas, siendo n el número de nodos o vértices del grafo.

Paso a paso en la solución de un grafo

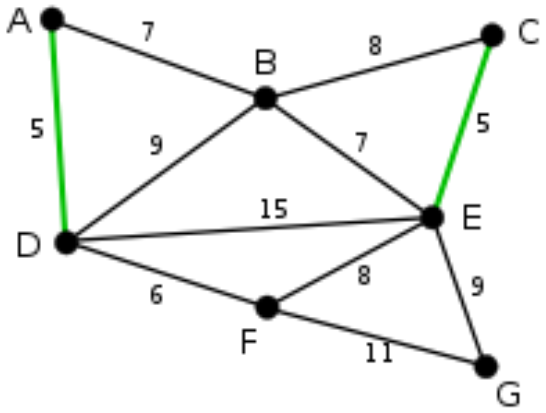


Este es el grafo inicial. Los números indican el peso de las aristas. Se ordenan las aristas de menor a mayor y se incia por la menor. En este caso hay dos aristas con el mínimo peso.

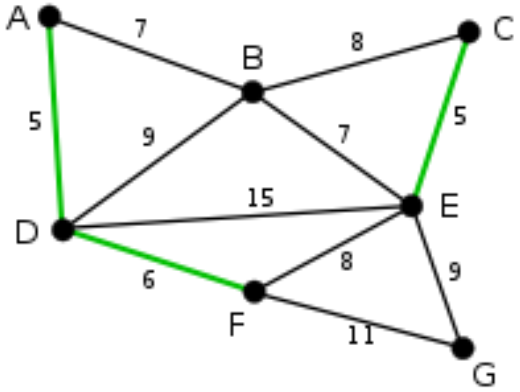


Se selecciona, en este caso de manera aleatoria la arista AD, como arista de inicio.

Paso a paso en la solución de un grafo

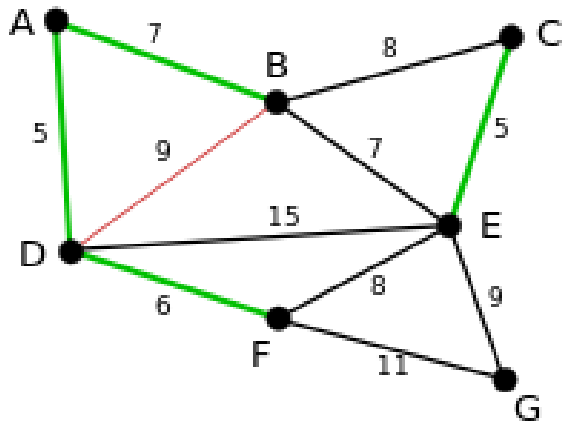


Se selecciona entre todas las aristas restantes, la de menor peso siempre que no forme un ciclo. En este caso la menor arista siguiente es CE.

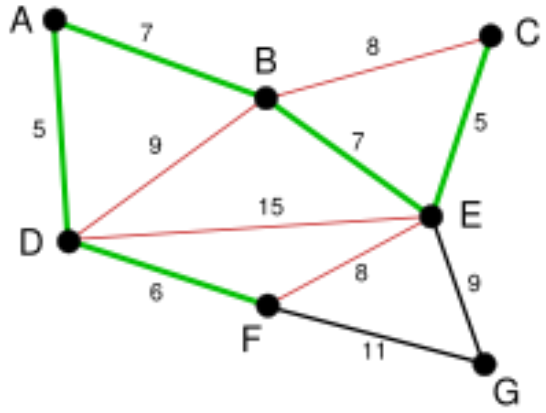


Seleccionamos DF, con peso 6, que es la siguiente arista de menor peso que no forma ciclos.

Paso a paso en la solución de un grafo

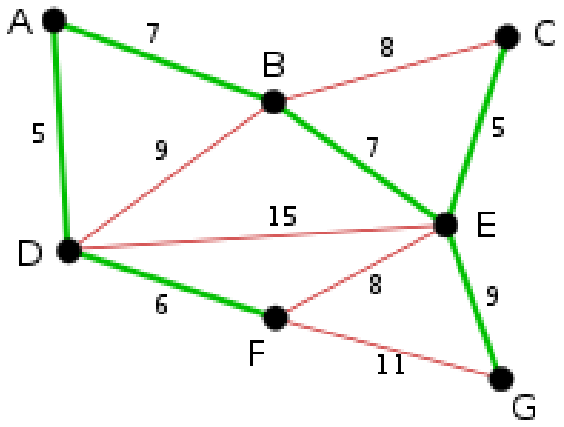


De las aristas restantes, las de menor peso son las aristas AB y BE, de peso 7. AB se elige aleatoriamente, y se añade al conjunto de las aristas seleccionadas. De este modo, la arista DB no puede ser seleccionada ya que formaría el ciclo ADB. Por tanto la marcamos en rojo.



Siguiendo el proceso seleccionamos la arista BE con peso 7. Además marcamos en rojo las aristas BC, DE y FE ya que formarían los ciclos BCE, DEBA, FEBAD respectivamente.

Paso a paso en la solución de un grafo



Por último, se selecciona la arista EG de peso 9. Como han sido seleccionadas un número de aristas igual al número de vértices menos uno, el proceso ha terminado. Se ha obtenido el árbol de expansión mínima con un peso de 39.

A fin de comprobar que el resultado es correcto hacemos la operación

Vértices = 7 Aristas = lvl-1 = 7-1 =6

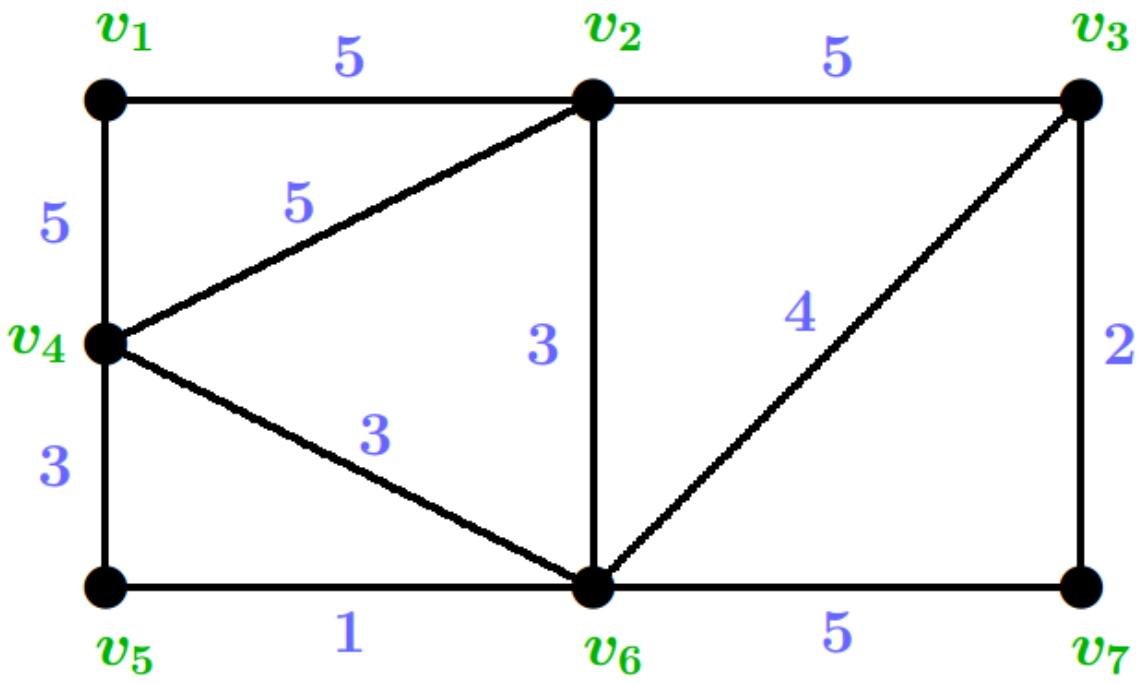
Aristas = 6

Peso del árbol = (5+7+6+5+9+7) = 39

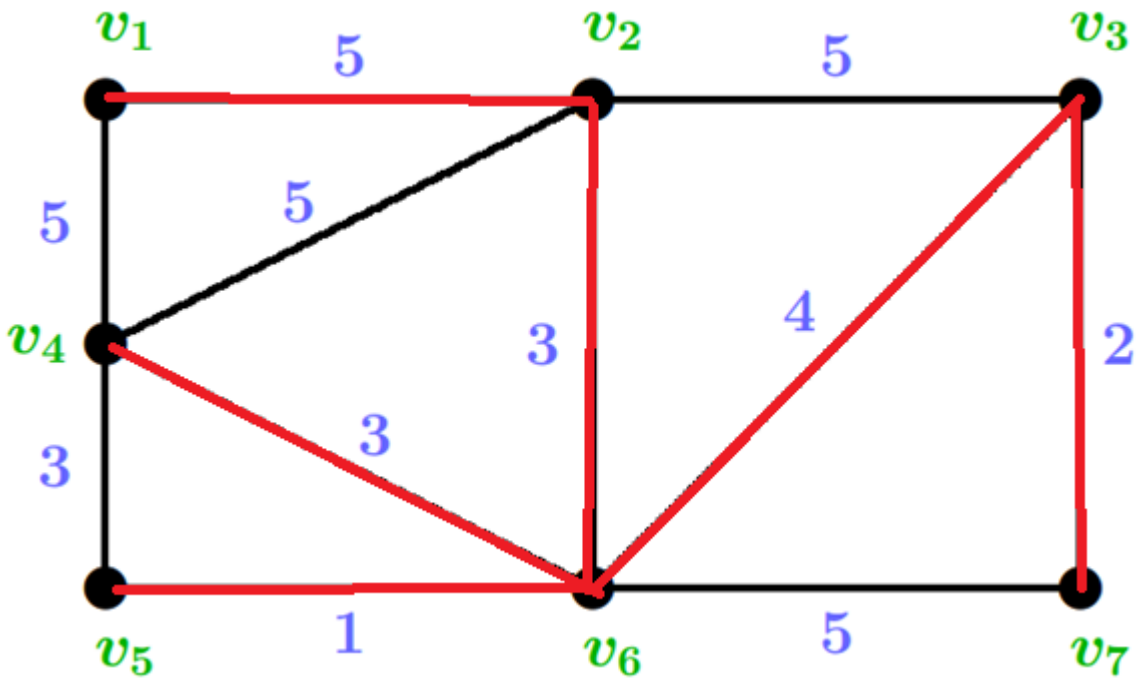
Pseudocódigo del algoritmo:

- Kruskal (G)
- E(1)=0, E(2)= todos los Arcos del grafo G
- **while** E(1) contenga menos de n-1 arcos y E(2)=0 **do**
- De los arcos de E(2) seleccionar el de menor costo -->e(ij)
- E(2)= E(2) - {e(ij)}
- **if** V(i), V(j) no están en el mismo árbol **then**
- juntar los árboles de V(i) y de V(j) en uno sólo
- end **if**
- end **do**
- Fin del algoritmo

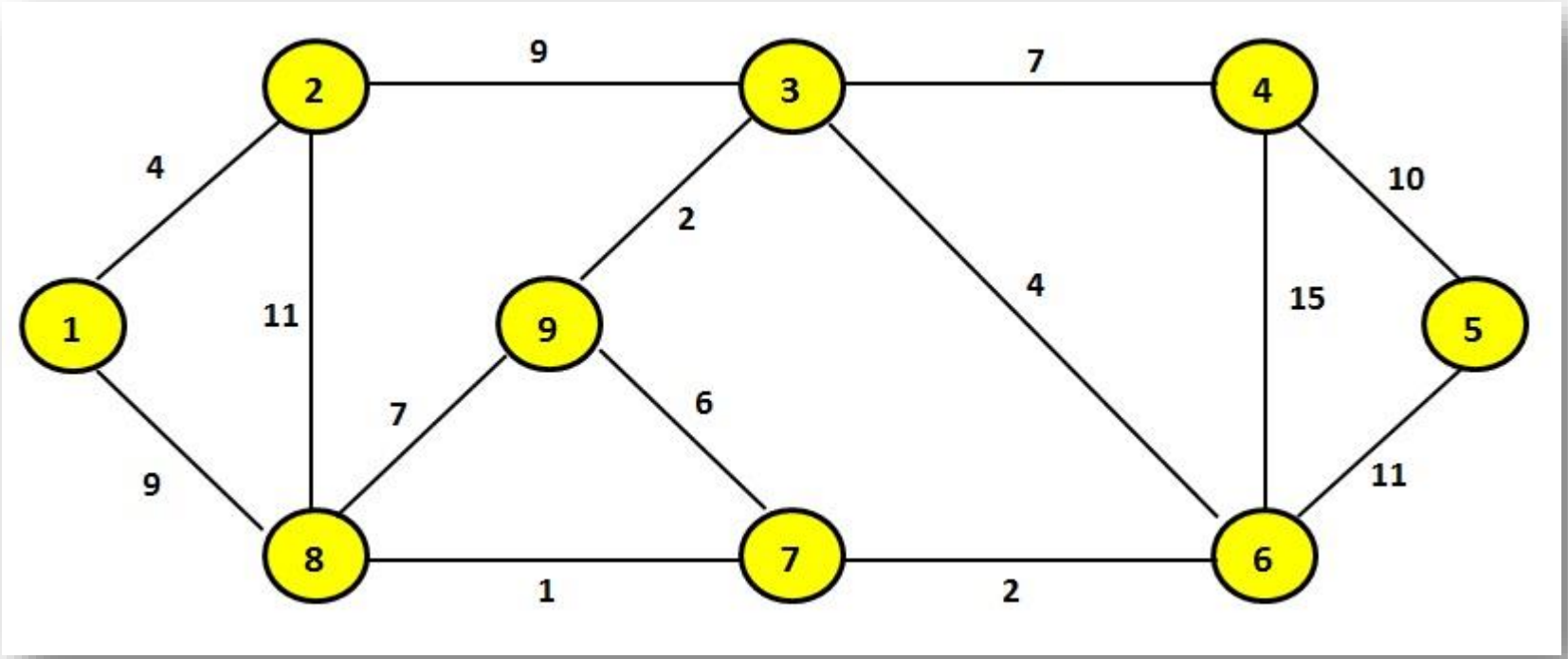
Ejemplo 1



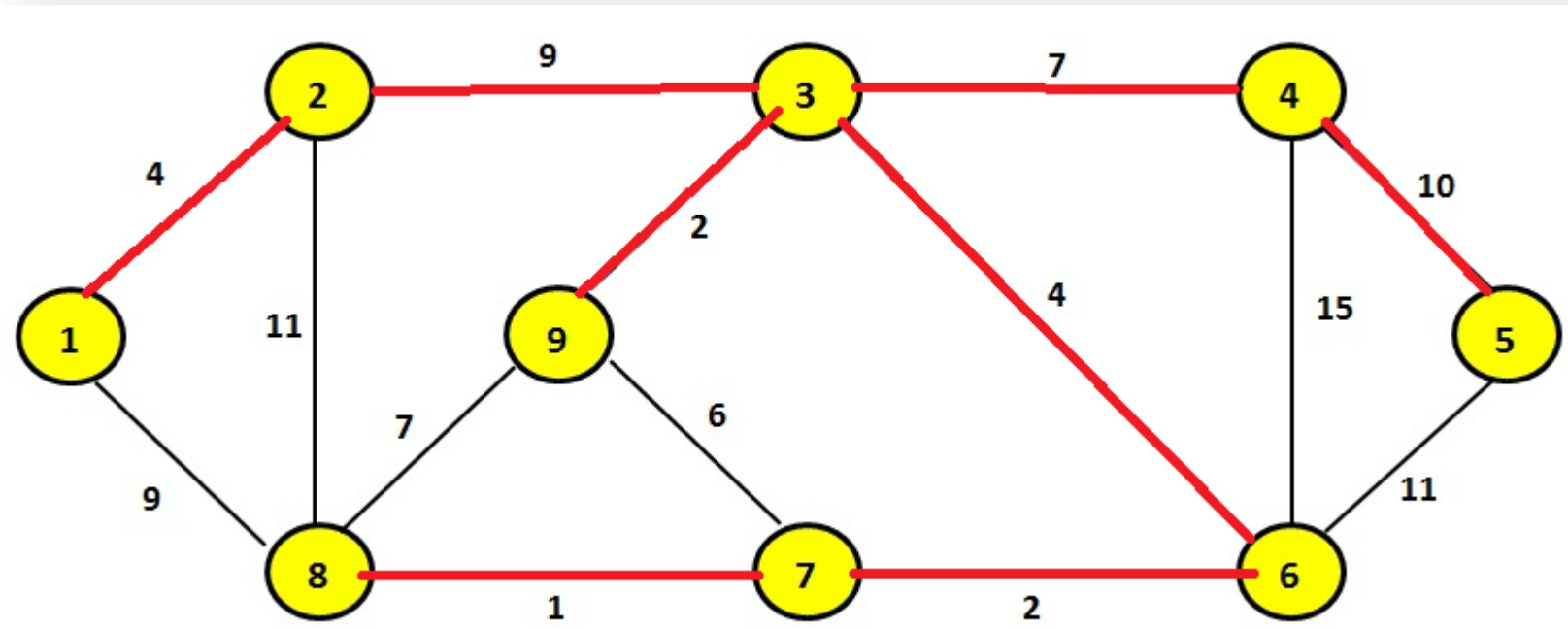
Solución



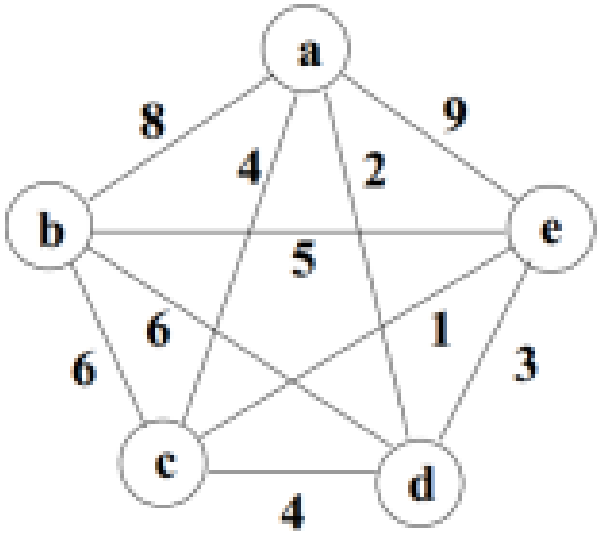
Ejemplo 2

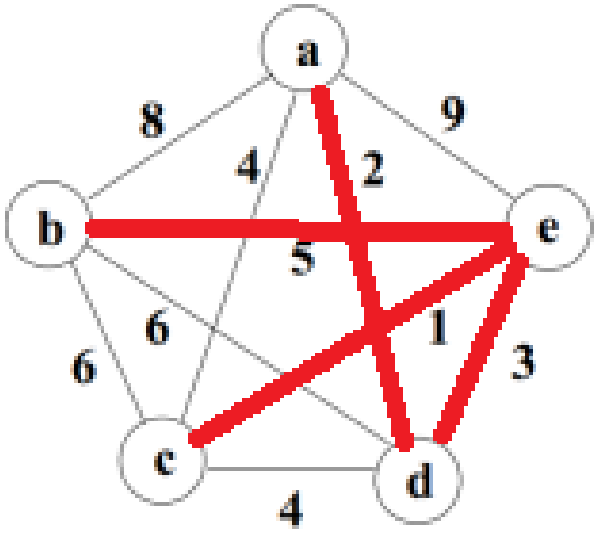


Solución



Ejemplo 3





Bibliografía

- https://es.wikipedia.org/wiki/Algoritmo_de_Kruskal
- https://sites.google.com/site/complejidadalgoritmicaes/kruskal
- http://arodrigu.webs.upv.es/grafos/doku.php?id=algoritmo_kruskal