

Anlisis de algoritmos

Primera entrega proyecto

Joaquin Suarez, *Estudiante, Pontificia Universidad Javeriana*

Resumen—Descripción de los problemas y de sus aspectos claves.

Palabras clave—mst, árbol de expansión, grafo, vértice, arista, algoritmo, kruskal, prim, DFS, BFS, búsqueda en profundidad, búsqueda en anchura, árbol de expansión mínimo



1 INTRODUCCIÓN

EN este artículo se tratarán 2 problemas de los planteados, para los cuales se realizar su respectivo análisis y explicación de los terminos claves que trata el mismo. Por ultimo se propondrán 2 algoritmos con los cuales se puede dar solucin al problema.

2 PROBLEMA 1

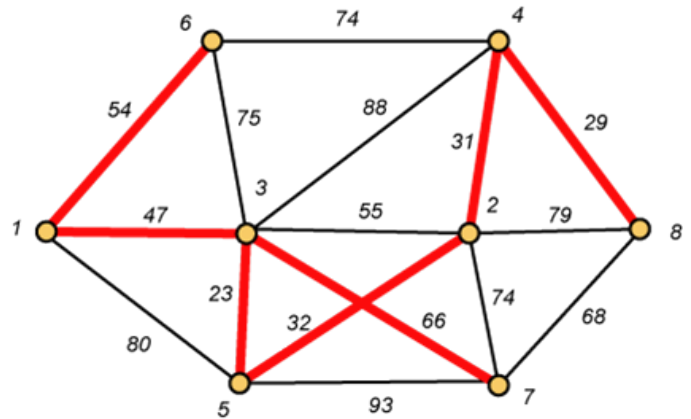
(MST). Dado un grafo $G = (V, E)$ con n vértices y m aristas. (El grafo podría representar una red telefónica). Cada arista es coloreada azul o roja. También esta dado un parámetro k como parte de la entrada. Proponga un algoritmo que encuentre un árbol de expansión sobre G con exactamente k aristas azules, y exactamente $n-k-1$ aristas rojas. Determine el tiempo de ejecucin del algoritmo y muestre que es correcto.

2.1 Conceptos claves

Grafo: La definición formal de grafo es "Un grafo G se define como un par (V, E) , donde V es un conjunto cuyos elementos son denominados vertices o nodos y E es un subconjunto de pares no ordenados que reciben el nombre de aristas o arcos"

Árbol de expansión minima: Es una ramificación que se genera recorriendo TODOS los vértices del grafo, tomando las aristas con menor valor y logrando así el recorrido de menor 'costo'. No se usan todas las aristas del grafo.

2.2 Ilustración



En la imagen vemos una posible instancia del problema que se trata, en el que no se pintan todas las aristas del grafo sino solo K aristas, el cual es un valor ingresado por el usuario, para el caso de la imagen vemos que el valor de K corresponde a 7, debido a que hay 7 aristas coloreadas a diferencia de las otras.

2.3 Solución

Para encontrar la solució a este problema podemos usar el algoritmo de busqueda en profundidad o Depth First Search, el cual recorre cada nodo desde la raiz y encuentra el arbol de recubrimiento del grafo, para aplicarlo al problema se podría modificar para recibir el parametro k y que de acuerdo al numero de aristas del grafo que recibe se coloreen de azul k aristas y exactamente $n-k-1$ aristas de rojo

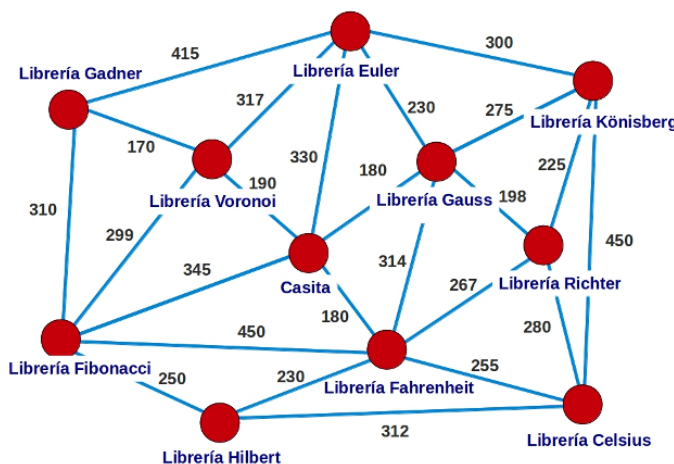
Otro algoritmo util para solucionar este problema es el de busqueda en anchura o Breadth First Search que al igual que la

solución anterior halla el árbol de expansión del grafo. También se debe modificar para que reciba el parámetro k y lleve una cuenta de cuantas aristas tiene el grafo, cuantas deben ser azules y cuantas rojas, que vaya disminuyendo a medida se vaya completando el árbol.

3 PROBLEMA 2

(LP) Suponga que se planea construir una nueva cadena de tiendas en una ciudad dada, usted tiene identificado una serie de ubicaciones potenciales en diferentes barrios. Además asuma que la demanda de productos en cada barrio de la ciudad es conocida. Si usted quiere construir exactamente k tiendas, ¿dónde debería localizarlas de forma que minimice la distancia promedio de los clientes? Si en lugar usted desea construir una cantidad variable de tiendas, y el costo de construir una tienda en cada sitio es conocido, ¿dónde debería construir las tiendas de forma que minimice el costo total del construccin y la distancia promedio de los clientes?

3.1 Ilustración



La imagen superior corresponde a una posible instancia del problema a tratar, vemos como tenemos una serie de ubicaciones potenciales en diferentes barrios para nuestra cadena de tiendas (que corresponde a cada librería en el diagrama) además sabemos que distancia hay entre cada punto (señalada por los números presentes en cada línea o arista que une a cada nodo de la gráfica).

3.2 Solución

Para llegar a la solución del problema podemos utilizar el Algoritmo de Kruskal, ya que este halla el árbol de expansión mínima mencionado en los términos claves del problema 1. Antes de ejecutar el algoritmo debemos construir el grafo con la cantidad k de nodos o tiendas que deseamos y ahí si podemos proceder a ejecutar el algoritmo sobre este grafo.

Otro algoritmo que puede ayudarnos a resolver el problema es el algoritmo de Prim, que también nos ayuda a encontrar el árbol de expansión mínima. Como en la primera solución, debemos construir primero el grafo con los k nodos o tiendas que deseamos y luego proceder a usar el algoritmo.

REFERENCIAS

- [1] UCI Dept. Information Computer Science, *Design and Analysis of Algorithms*, Lecture notes for February 6, 1996. Disponible <http://www.ics.uci.edu/~eppstein/161/960206.html>
- [2] Geeks for Geeks *Greedy Algorithms — Set 2 (Kruskals Minimum Spanning Tree Algorithm)* Aashish Barnwal, reviewed by Geeks for Geeks team, Disponible <http://www.geeksforgeeks.org/greedy-algorithms-set-2-kruskals-minimum-spanning-tree-mst/>
- [3] Geeks for Geeks *Greedy Algorithms — Set 5 (Prims Minimum Spanning Tree (MST))* Geeks for Geeks team, Disponible <http://www.geeksforgeeks.org/greedy-algorithms-set-5-prim-s-minimum-spanning-tree-mst-2/>
- [4] Wikipedia *Busqueda en profundidad* Disponible https://es.wikipedia.org/wiki/B%C3%BAsqueda_en_profundidad
- [5] UdeA 3.2. *Busqueda en Profundidad. Busqueda en anchura.* Universidad de Antioquia, Disponible http://docencia.udea.edu.co/regionalizacion/teoriaderedes/informaci%C3%BAn/C3_Profundidad.pdf