Árbol Generador Mínimo

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Algoritmos y Estructuras de Datos III

Repaso de definiciones

Definición (Árbol generador)

Sea G un grafo. Decimos que T es un *árbol generador* de G si se cumplen estas condiciones:

- T es subgrafo de G.
- T es un árbol.
- T tiene todos los vértices de G.

Definición (Costo de un árbol generador)

Sea G un grafo con pesos en sus aristas. Si T es un árbol generador de G, definimos el *costo* de T como la suma de los pesos de las aristas de T.

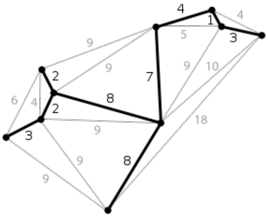
¿Qué es un AGM?

Definición (AGM)

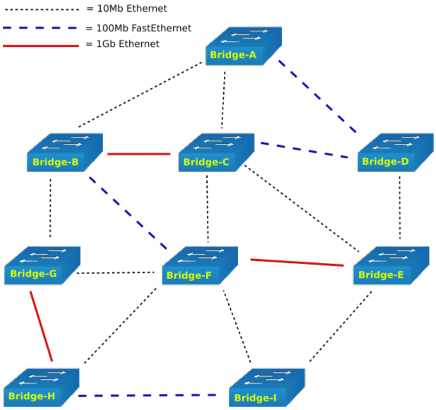
Sea G un grafo. Decimos que un grafo T es un árbol generador mínimo (AGM) de G si:

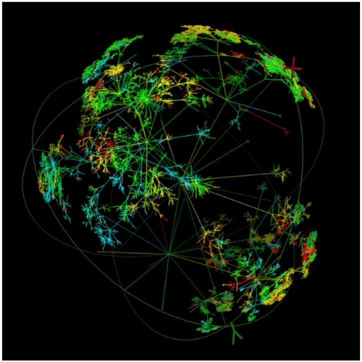
- T es un árbol generador de G.
- El costo de T es mínimo con respecto a todos los árboles generadores de G.

Aca vemos un grafo G (en gris) y un subgrafo T (en negro) que es AGM de G.

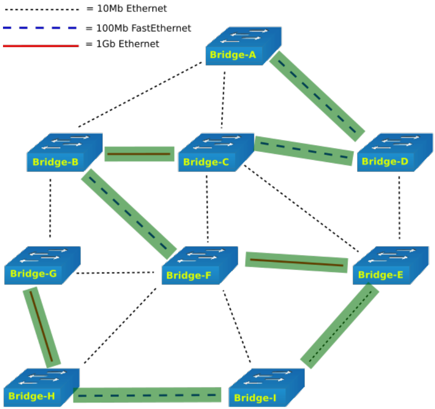


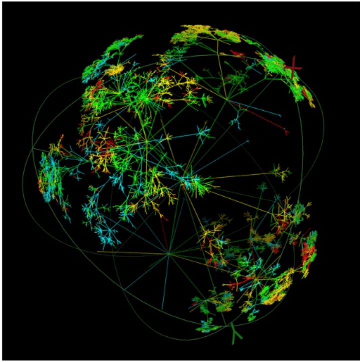
Spanning Tree Protocol (STP)





Spanning Tree Protocol (STP)





Arturo y las chinchillas

Arturo se compró una mansión en la cordillera de los Andes con el objetivo de estudiar a las chinchillas. Las chinchillas viven en tuneles subterráneos y cada una tiene un circuito de túneles por el que se mueve. Arturo quiere colocar camaras en los túneles de manera de estudiar el movimiento de las chinchillas. Para ello cuenta con un mapa de los túneles subterráneos que consiste de puntos de intersección entre túneles y la longitud entre dos puntos de intersección.
Sabe que colocar una cámara en un túnel de longitud l le cuesta l pesos y que cada chinchilla se mueve por un ciclo de túneles cerrados sin pasar dos veces por el mismo punto hasta volver al inicio del ciclo. Quiere colocar las cámaras de maneras de que todo ciclo posible de alguna chinchilla posea una cámara y gastando la menor cantidad de dinero.
Quiere que le digamos cuanto la va a salir realizar esto y cual es el costo de la cámara más cara que tiene que colocar.

Especificaciones extra

- Para esta clase podemos suponer que el grafo de túneles es conexo (pensar qué cambiaría en la solución si no lo es) y que tiene al menos un ciclo.
- El formato del input es una linea con V y E, la cantidad de vértices y aristas del grafo de túneles, seguido de E lineas con tres enteros a_i, b_i y c_i (separados por espacios) que representan que hay un túnel entre los vértices a_i y b_i con costo c_i.
- Los vértices están numerados de 0 a $V - 1$.
- Debemos imprimir dos enteros C y L (separados por espacios) que son el costo mínimo total y el costo de la cámara más cara que pusimos.
- El algoritmo deberá tener una complejidad temporal de $O(V + ElogV)$.

Sugerencia: Usar AGM.

Definición 1

Si S es un subgrafo de G decimos que es *solución* del problema para G si todo ciclo simple de G tiene una arista de S

Definición 1
Si S es un subgrafo de G decimos que es *solución* del problema para G si todo ciclo simple de G tiene una arista de S

Definición 2
Si S es un subgrafo de G llamamos $G - S$ al subgrafo de G que tiene los vértices de G y las aristas de G que no están en S.

Definición 1
Si S es un subgrafo de G decimos que es *solución* del problema para G si todo ciclo simple de G tiene una arista de S

Definición 2
Si S es un subgrafo de G llamamos $G - S$ al subgrafo de G que tiene los vértices de G y las aristas de G que no están en S.

Observación 1
S es una solución minimal para el grafo G si y sólo si $G - S$ es un árbol generador.

Observación 2

S es una solución de costo mínimo para el grafo G si y sólo si $G - S$ es un árbol generador de costo máximo de G.

Observación 2

S es una solución de costo mínimo para el grafo G si y sólo si $G - S$ es un árbol generador de costo máximo de G.

Observación 3

El algoritmo de Kruskal calcula un árbol generador de costo máximo si ordenamos las aristas de mayor a menor costo (en lugar de menor a mayor).

Observación 2

S es una solución de costo mínimo para el grafo G si y sólo si $G - S$ es un árbol generador de costo máximo de G.

Observación 3

El algoritmo de Kruskal calcula un árbol generador de costo máximo si ordenamos las aristas de mayor a menor costo (en lugar de menor a mayor).

Observación 4

La arista más pesada es la primera que no agregamos a la solución en el algoritmo de Kruskal.

Problema[1]

- En un país con n ciudades. La i-ésima ciudad tiene coordenadas (x_i, y_i). Ninguna de ellas tiene electricidad.
- Hay dos maneras de proveer electricidad a una ciudad:
  1. Construir en ella una planta de energía.
  2. Tirar un cable entre esta ciudad y otra que ya tenga electricidad. Para unir ciudades i y j necesitamos $|x_i - x_j| + |y_i - y_j|$ kilómetros de cable.
- Sabemos que construir una planta cuesta P y un kilómetro de cable, K.
- ¿Cuál es la manera más barata de que todas las ciudades tengan acceso a la electricidad?

[1] `https://codeforces.com/problemset/problem/1245/D`

Una solución

- Crear un grafo con un vértice correspondiente a cada ciudad y un vértice adicional.
- Agregar aristas entre las ciudades con el costo de unir las ciudades con cable.
- Agregar aristas entre cada ciudad y el vértice adicional con costo equivalente a instalar una planta de energía en la ciudad.
- Un AGM en este grafo respresenta una solución al problema.

Hiperconectados[2]

Tenemos un conjunto de *n* ciudades, para las cuales queremos crear una red de conexiónes que cumpla que todo par de ciudades está conectado por al menos un camino de nuestra red. Para lograr esto, tenemos *m* potenciales conexiones entre pares de ciudades. Cada potencial conexión posee tres atributos: dos ciudades *a* y *b*, que indica simétricamente que ciudades serían conectadas y *c* que representa el costo de instalar dicha conexión. Nuestro objetivo es crear la red indicada de manera tal que el costo total de la misma sea el mínimo, para lo cual ya sabemos que existen algoritmos que nos pueden ayudar específicamente con dicha tarea. Sin embargo, nuestro problema no se reduce simplemente a esto. Si bien las conexiones tienen un valor asociado que nos indica cuanto nos costará su instalación, hay otras variables asociadas a las conexiones que se están ocultando por una cuestión de simplicidad. Una conexión podría tener un costo de instalación bajo, pero a la vez podría ser una conexión no muy deseada por encontrarse en una zona sísmica, lo cual aumentaría las chances de tener que ser sometida a reparaciones. También podría haber una conexión con costo asociado bajo, pero que su instalación genere un descontento público por el lugar donde debe realizarse la instalación.
Es por estas situaciones que más alla de encontrar la forma menos costosa de armar una red entre las *n* ciudades tal que cada par de ciudades se encuentre conectado de al menos una manera (situación que denominaremos Anexión General Minúscula o AGM por sus siglas), tendremos aparte el siguiente problema. Por nuestros conocimientos en el área sabemos que la solución AGM del problema no tiene porque ser única.

Queremos saber entonces por cada posible conexión si la misma tiene la propiedad de 1) Estar en todo AGM del problema 2) No estar en ningún AGM del problema 3) Estar en algún AGM del problema.

[2] `https://codeforces.com/contest/160/problem/D`

Solución

- El orden de las aristas luego de ordenarlas con Kruskal.
- Procesamos en lote a todas las aristas que tengan el mismo costo.
- Armar un grafo H con vértices representados por componentes conexas y las aristas son el lote de aristas que tienen el mismo costo.
- Si una arista es un puente H, dicha arista pertenece a todos los AGM.
- Si una arista une a un vértice con él mismo en H, no pertenece a ningún AGM.
- Si no pertenece a algún AGM.