Problema 2 - La extraña aventura de Darío

Autor: Josué Rodríguez Ramírez

Análisis del Problema:

Darío estaba comiendo unos champiñones cuando de pronto se quedó dormido. Al despertar, se dio cuenta de que ya no se encontraba en la tierra. Estaba en un extraño planeta rodeado de hombrecitos verdes de abundante cabellera y etremidades cortas. El representante de los extraterrestres se acercó a Darío y le explicó la situación. Aparentemente los hombrecillos tenían tecnología lo suficientemente avanzada para conectar planetas por toda la galaxia con portales de traslación instantánea, pero nunca se les había dado bien la matemática discreta (al parecer una pandemia les dificultó el aprendizaje), así que abdujeron a Darío, estudiante de MATCOM para pedirle ayuda.

El problema es el siguiente: Tienen n planetas unidos por m portales. Los portales van en una sola dirección y el costo de combustible espacial de cada uno (w_i) es conocido. Se sabe que sólo un portal puede conectar dos planetas. Es posible escoger un conjunto de portales e invertir el sentido de todos al mismo tiempo. El costo de realizar esa acción está dado por el peso máximo entre todas las portales que fueron invertidas. Los extraterrestres quieren encontrar una forma de, a partir de esas inversiones de portales, conseguir que haya un planeta al que sea posible llegar desde cualquier otro planeta. Ayude a Darío a encontrar una forma de conseguir esto, con costo mínimo.

Explicación del problema

El problema planteado se puede simplificar usando teoría de grafos:

Se tiene un grafo dirigido y ponderado, donde el peso de cada arista es el costo de invertirla. Se quiere encontrar un nodo al que se pueda llegar desde cualquier otro nodo, invirtiendo aristas, con el costo mínimo, que está dado por la arista de mayor peso.

Para la solución tenemos que preguntarnos varias cosas:

- ¿Cómo saber qué aristas invertir?
- ¿Cómo probar que puedo llegar desde cualquier nodo a un nodo v específico?

Solución del problema

Deseamos invertir las aristas de tal manera que obtengamos el menor costo necesario para cumplir la exigencia del problema. Pudiéramos recorrer los pesos de las aristas desde el menor hasta el mayor e ir probando invirtiendo las aristas menor o igual a ese valor hasta obtener el costo mínimo que en el que se logre encontrar el nodo v.

Pero nos podemos dar cuenta de algo, iremos probando pesos hasta que uno de ellos lo cumpla (a partir de ahí todos sobre-cumplirán) porque estamos recorriendo de menor a mayor los pesos. Esto nos permite reconocer que pudiéramos hacer BÚSQUEDA BINARIA sobre los pesos de las aristas. Y esta es la primera solución:

> Tomamos el peso maximo de las aristas y hacemos búsqueda binaria de 0 a ese máximo probando si al invertir las aristas menores del valor de la búsqueda obtenemos el nodo v. Si es así, continuamos la búsqueda con la mitad inferior, y en caso contrario, la mitad superior. Devolvemos el menor valor de la búsqueda binaria que cumpla la condición, la cual analizaremos ahora (aunque tenemos un problema con esta solución y por lo tanto propondemos una mejora más adelante)

Vamos a responder la segunda pregunta, que trata de esa condición que preguntamos en la búsqueda binaria (y que es la misma en ambas condiciones):

Sería muy ineficiente probar por cada nodo v si es posible llegar hasta él desde todos los demás. Pero esto se traduce en si se puede llegar desde v a todos los demás, solo necesitamos que las aristas estén invertidas. Fijémonos que si hay un camino v $\rightarrow$ w al invertir las aristas obtenemos un camino w $\rightarrow$ v. Entonces podemos asegurar que si hay un camino de v a todos los nodos, entonces existe una manera de llegar desde todos los nodos hasta v.

Para esto haremos un DFS, para saber si se pueden visitar todos los nodos desde v.

Respondidas las dos preguntas, obviamente nos surge otra interrogante:

- ¿Cómo unir estos dos resultados para obtener la solución del problema?

Si tenemos las aristas que debemos invertir y la manera de comprobar que se llega a v desde todos los nodos podemos hacer un grafo alternativo al que recibimos. Este nuevo grafo tendrá las mismas aristas que el grafo original pero insertando las aristas que tiene un costo menor que el k tomado de la busqueda binaria con sentido contrario. Es decir, si la arista v $\rightarrow$ w tiene costo menor que k en el grafo alternativo tendremos ambas aristas (v $\rightarrow$ w y w $\rightarrow$ v).

Con esta estrategia podemos invertir a la vez todas las aristas menores que k (ya que el costo el maximo, es decir k), si no era necesario invertir todas no importa porque el camino permanece intacto al mantener la arista original. Si quitamos las aristas y nos quedamos con las invertidas solamente entonces tendríamos un error, responderíamos un problema distinto.

Ahora sí, podemos unir al segundo resultado lo que tenemos, en este grafo alternativo invertimos las aristas (si nos damos cuenta solo debemos invertir las arista mayores que k) y realizamos dfs en el grafo. Si encontramos el nodo v que alcanza todos los demás nodos devolvemos $True$ y sabemos que corremos la búsqueda binaria a la mitad inferior, si devolvemos $False$ será en la mitad superior.

Correctitud

Aunque anteriormente estuvimos probando algunos resultados, nos quedan algunas cosas por agregar:

El algoritmo termina porque la búsqueda binaria se realiza sobre un conjunto finito de elementos y en cada paso se analiza con un dfs si nodos de un grafo han sido visitados. Como la búsqueda binaria reduce el espacio de búsqueda, entonces se llega a comprobar, por PBO, que acaba.

Todos los elementos que la búsqueda binaria desecha no son parte de la solución porque, o bien son menores que un valor k_1 el cual no fue solución y por lo tanto, en menos costo no se puede resolver. Esto sucede porque al invertir todas las aristas menores o iguales que k_1 probamos si existe un subconjunto de ellas con las que se puede resolver el problema. O bien son mayores que un valor k_2 y como este valor es menor que los demás, no es necesario comprobar otros valores que cumplirán la condición en un costo mayor.

Veamos que el resultado obtenido es el que queremos:

Tenemos el valor de costo k e insertamos las artistas con costo menor o igual que ese valor. Estas aristas pudieran ser parte de los caminos de todos los vertices hasta v, o bien desde v a todos los vértices si hablamos del grafo alternativo. Supongamos que el algoritmo nos da un costo mayor al que realmente es. Entonces tenemos que notar que la busqueda binaria probará la mitad inferior a ese costo, y eventualmente, el costo que debería ser. Y devolverá el menor de los costos, por tanto, ese que realmente es solución. De manera análoga ocurre al suponer que dé un resultado menor.

Optimizando la solución:

Un problema con esta solución es que los pesos (como vimos en un ejercicio similar en clase practica) pueden tener valores muy grandes y por tanto la complejidad temporal aumentaría mucho.

¿Cómo solucionar esto?

Si ordenamos las aristas por los pesos y hacemos búsqueda binaria en las aristas entonces la complejidad temporal dependería solo de la cantidad de nodos y aristas y no en los pesos.

La correctitud es igual que la solución anterior, solo que esta busca por pesos que las aristas ya presentan.

Complejidad temporal

Como realizamos DFS por cada uno de los pesos de la búsqueda binaria entonces, para la primera solución tenemos:

> $O((|V| + |E|) * log(k))$ donde $|V|$ es la cantidad de vertices $|E|$ es la cantidad de aristas y k peso máximo de las aristas. Notar que $|v| + |E|$ es el costo del DFS.

Para la mejora del algoritmo, hicimos la búsqueda en el espacio de las aristas, por tanto:

> $O((|v| + |E|) * log(E))$