Explicación Práctica: Almost Shortest Path

Santiago Alessandri

September 10, 2011

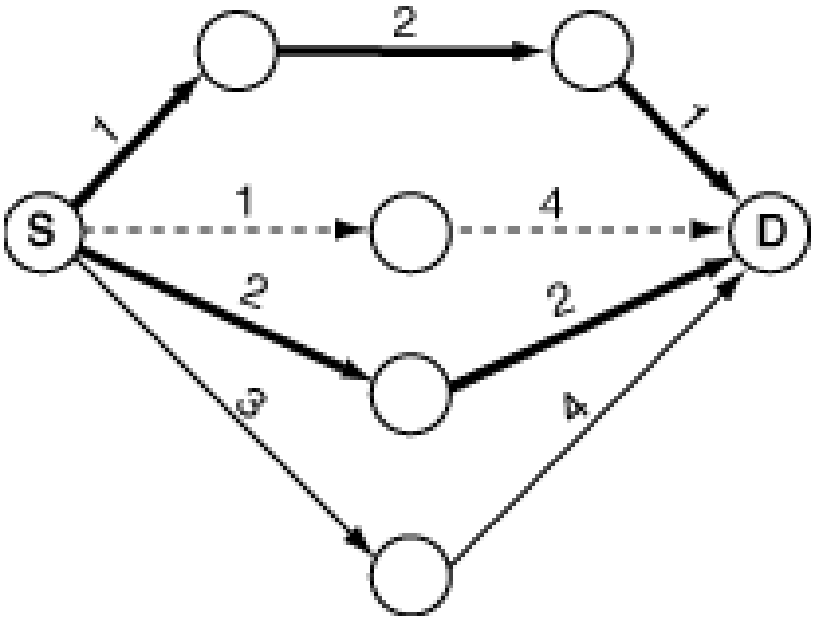
Enunciado

Finding the shortest path that goes from a starting point to a destination point given a set of points and route lengths connecting them is an already well known problem, and it's even part of our daily lives, as shortest path programs are widely available nowadays.

Your program must answer not the shortest path, but the almost shortest path. The almost shortest path is defined as the shortest path that goes from a starting point to a destination point such that no route between two consecutive points belongs to any shortest path from the starting point to the destination. There could exist no possible answer as well.

Objetivo

¿Qué piden que se haga?
Si tenemos el siguiente grafo:



El camino mínimo tiene un costo de 4. Hay 2 y están en **negrita**.
El "Almost Shrotest Path" es el que esta interlineado y tiene un peso de 5, esa es la respuesta correcta.

Razonando el problema

¿Qué algoritmo utilizarían para determinar el camino mínimo?

- Grafo pesado.
- Grafo dirigido.
- No contiene aristas negativas.

$\rightarrow$ **Algorítmo de Dijkstra**.

¿Alguna posible solución?

¿Cuál es la clave de este problema?

Cuestión Clave

Aquí todo el problema consiste en como identificar las aristas que pertenecen a algún camino mínimo.

Si uno encuentra una forma eficiente de identificarlas, lo único que queda hacer es eliminarlas y luego ejecutar un *Dijkstra* sobre el grafo sin dichas aristas para obtener el *Almost shortest path* si es que lo hay.

¿Se les ocurre como hacerlo?

Identificando las Aristas

Teniendo:

- $S =$ vértice origen.
- $D =$ vértice destino.
- $(u, v, w) =$ arista que une los vértices u y v con costo w.
- $mp(u, v) =$ camino mínimo de u a v.

Si queremos determinar si (u, v, w) pertenece a algún camino mínimo de S a D, ¿cómo podemos hacer?

$$(u, v, w) \in mp(S, D) \iff mp(S, u) + w + mp(v, D) = mp(S, D)$$

Podemos validar esta condición para cada (u, v, w) del grafo original y eliminar las aristas donde la condición es verdadera.

¿Qué necesitamos?

Costos Necesarios

Para poder validar la condición de la diapositiva anterior es necesario tener los siguientes elementos:

- Valor de w. Este elemento lo tenemos en cada arista.
- Valor de $mp(S, u)$, $\forall u \in G$.
- Valor de $mp(v, D)$, $\forall v \in G$.
- Valor de $mp(S, D)$. Esto queda implícito al calcular el segundo elemento de la lista.

Utilizando *Dijkstra* podemos obtener facilmente los $mp(S, u)$. Ya que *Dijkstra* nos da el camino mínimo de un vértice a todos los demás.

¿Cómo se puede obtener $mp(v, D)$?

Obteniendo todos los $mp(v, D)$

Se podría plantear que para obtener los caminos mínimos de todos los vertices a D:

- Utilizar el algoritmo de *Floyd-Warshall* y de esta forma obtener el camino mínimo entre todo par de vertices, tanto los $mp(S, u)$ como los $mp(v, D)$. Pero el costo de este algoritmo es $O(|V|^3)$, lo cual es un costo muy alto.
- Utilizar el algoritmo de *Dijkstra* sobre cada vértice del grafo. Esto también lleva un costo muy alto con $O(|V||V|^2) = O(|V|^3)$ en caso de usar listas o $O(|V||E|log|V|)$ utilizando *heap*.

Ambas soluciones son inviables en cuanto a tiempo de ejecución.

Obteniendo todos los $mp(v, D)$

El camino mínimo de v a D en el grafo G es una secuencia de aristas:
$(v, x_i)(x_i, x_{i+1})...(x_n, D)$

Entonces en el grafo G', que es G invertido(un grafo invertido es aquél que tiene los mismos vértices pero cuyas aristas han cambiado de dirección), el camino mínimo de D a v será:
$(D, x_n)...(x_{i+1}, x_i)(x_i, v)$

$\rightarrow$ Si lanzamos *Dijkstra* desde D en G' obtenemos el camino mínimo de D a todos los vertices que en G son los caminos mínimos de todos a D. Obteniendo asi $mp(v, D)$, $\forall v \in G$

Programando la Solución

Para cada *test case* nuestro algoritmo hara lo siguiente:

1. Genera G y G' en simultaneo. Esto puede hacerse debido a que los límites de tamaño del grafo son relativamente pequeños y el costo de memoria no es significativo.
2. Lanza un *Dijkstra* desde S sobre G obteniendo todos los $mp(S, u)$.
3. Lanza un *Dijkstra* desde D sobre G' obteniendo todos los $mp(v, D)$.
4. Elimina las aristas de G que cumplan con la condición vista.
5. Lanza un *Dijkstra* desde S sobre G obteniendo el *Almost Shortest Path*, si existe.

Orden de ejecución de la solución

Para calcular el orden del tiempo de ejecución hay que calcular el mismo para cada parte:

1. $O(|V| + |E|)$ o $O(|V|^2)$ (depende implementación)
2. $O(|E|log(|V|))$ o $O(|V|^2)$ (depende implementación)
3. $O(|E|log(|V|))$ o $O(|V|^2)$ (depende implementación)
4. $O(|E|)$
5. $O(|E|log(|V|))$ o $O(|V|^2)$ (depende implementación)

$\rightarrow O(|V|^2)$ en el peor de los casos. Esto es un tiempo razonable.

Conclusion

Como se puede ver en este problema, muchas veces los algoritmos que se necesitan conocer para resolver el problema son algoritmos tradicionales.

La clave en este tipo de problemas es saber como combinarlos para lograr el objetivo deseado.

La presentación y el código fuente los pueden descargar desde:

- `http://wiki.san-ss.com.ar/acm-icpc-4210`