ELIGIENDO EL HORARIO

AUTOR: LAUTARO LASORSA

Solución Esperada

Para este problema existen 2 soluciones esperadas, de las cuales se adjunta además una posible implementación.

Una observación común a ambas soluciones es el problema presenta un grafo, donde las paradas son los nodos y los colectivos las aristas.

Solución Esperada 1

Está solución consiste en enfocar el problema al revés, es decir, partir del nodo de destino en el último momento permitido, e intentar llegar a al nodo de origen en el horario más tarde posible.

Es decir, para cada nodo responder "¿Cuál es el último momento en el que si estoy en este nodo, puedo llegar al destino en el tiempo permitido?"

La forma es así:

Si un colectivo sale de una parada **A** en un tiempo $\mathbf{T_0}$ y llegar a una parada **B** en $\mathbf{T_1}$, entonces voy a pensar que me permite salir de la parada **B** en $\mathbf{T_1}$ y llegar a **A** en $\mathbf{T_0}$.

Entonces, en cada parada (nodo del grafo) voy a tomar mayor tiempo en el que puedo estar ahí. Entonces, me van a interesar todos los colectivos que me hayan podido llevar ahí en un momento igual o anterior. Al máximo momento en el que podemos llegar al nodo lo llamaremos tn

Por esto, para cada colectivo **i**, se que me permite llegar a donde estoy en $\mathbf{I_i}+\mathbf{T_i}$ y sale de su origen en Ii, por tanto si el $\mathbf{t_n}<\mathbf{I_i}+\mathbf{T_i}$, es imposible que usando ese colectivo lleguemos al actual nodo y después podamos llegar al final al tiempo.

Si es posible usar ese colectivo (arista del grafo) para llegar al actual nodo, entonces nos va a interesar de sus repeticiones la última que nos es útil. Por tanto, vamos a mirar $\mathbf{d} = \mathbf{t_n}-(\mathbf{I_i}+\mathbf{T_i})$, es decir, la diferencia entre la primera vez que llega el colectivo y el último momento donde nos sirve.

Sabemos que podemos retrasar el colectivo cualquier múltiplo de $\mathbf{F_i}$ (no tomándolo cuando pase), por tanto buscaremos el mayor múltiplo de $\mathbf{F_i}$ menor o igual a **d**, llamémosle **f**, y entonces el mejor tiempo para ese colectivo es I_i+$\mathbf{T_i}$+**f**, por lo que podemos salir de su destino en $\mathbf{I_i}$+**f**.

Con esto, podemos aplicar el algoritmo de **Dijkstra** (tener presente que en este caso debemos mirar primero los nodos con mayor $\mathbf{t_n}$) y obtener el $\mathbf{t_0}$ en complejidad **(N+M)*log(N+M)**.

Solución Esperada 2

En esta solución hay 2 observaciones fundamentales:

- Si elijo el momento en el que sale, puedo determinar cuál es el mínimo momento al que puede llegar.
- Si sale más tarde, llegaré al mismo tiempo o después.

Por tanto, lo que debemos hacer es elegir el momento en el que salir de la casa. Luego hay que ejecutar el algoritmo de **Dijkstra** de forma usual. En esta caso, al llegar a una parada (nodo del grafo) podremos tomar un colectivo en el primer momento que pase que sea igual o mayor al momento en el que llegamos.

Así, el si mejor momento para llegar al nodo tn, del que parte el colectivo **i**. Si **tn <=** $\mathbf{I_i}$, podrá tomarlo en Ii, sino, llamemos **d =** $\mathbf{t_n-I_i}$, buscaremos el primer múltiplo de $\mathbf{F_i}$ **>= d**. Así, usando ese medio se llega a $\mathbf{B_i}$ en $\mathbf{I_i}$+$\mathbf{F_i}$+$\mathbf{T_i}$.

Con esto, podemos hacer una **busqueda binaria** en el momento en el que Gastón sale de su casa.

En este caso, la complejidad queda **(N+M)*log(N)*log(llegada)**

Soluciones Parciales

En las primeras subtarea el valor de llegada está acotado, de tal forma que se puede hacer algo similar a la **Solución Esperada 2**, pero probando todos los posibles momentos de salida.

En la primer subtarea se acota además el tamaño del grafo, por lo cuál puede probar manualmente cuál es la primera pasada de dicho colectivo que le sirve, y/o descomponer cada nodo en función del tiempo de llegada a dicho nodo.

Así, la complejidad esperada queda **llegada^2*(N+M)** para la subtarea 1, y **llegada*(N+M)** para la subtarea 2.

En la tercer subtarea el grafo es 1un árbol, por lo cual no es necesario implementar el algoritmo de Dijkstra sino que se pueden hacer algoritmos más sencillos sobre árboles (aprovechando que hay un único camino entre cada par de nodos). La complejidad en este caso es **N+M** o bien **log(llegada)*(N+M)**.

En la cuarta subtarea lo que se hace es que el valor de $\mathbf{F_i}$ sea 1 para simplificar las cuentas relativas a calcular el próximo colectivo que conviene tomar / del colectivo que conviene haber partido. Por lo demás se espera que sea igual al caso general.