

GPS anticongestión

Contribución de Agustín Santiago Gutiérrez

Descripción del problema

Los “GPS” son dispositivos electrónicos que aprovechan señales satelitales para calcular la posición actual del vehículo en el que se los utiliza. Es por esto que son utilizados por los conductores para determinar el camino hacia su destino, dado que el dispositivo indica en todo momento la calle por la que se debe continuar el viaje.

Estos dispositivos suelen indicar el camino más rápido posible para que los usuarios estén contentos y así aumentar las ventas. Pero tanta comodidad y conveniencia trajo su propio problema: como son muchos los usuarios que utilizan estos GPS, el camino más rápido propuesto comienza a congestionarse ¡y entonces el tiempo de viaje termina siendo peor!

Para resolver esto y superar a los competidores, la renombrada empresa OIA (*Orientación e Indicaciones Automovilísticas*) ha decidido sacar al mercado un *GPS anticongestión*. Este dispositivo no sólo indicará un camino óptimo, sino que además permitirá al usuario seleccionar **los K mejores caminos existentes**. La idea es que de esa forma se pueda tomar otro buen camino para evitar la congestión.

Tu tarea consiste en escribir una función que calcule los K mejores tiempos posibles de viaje, correspondientes a los K mejores caminos posibles. No es necesario calcular los caminos en sí, ya que de eso se encargarán futuros desarrolladores.

Aclaración: Un camino es una secuencia de calles utilizadas que podría contener calles repetidas. Dos caminos se consideran diferentes a menos que sus secuencias de calles sean completamente idénticas.

Descripción de la función

Debes implementar la función `gps(N,M,comienzo, fin,K : ENTEROS; origen, destino, tiempo : ARREGLO[M] de ENTEROS)`

Donde N es la cantidad de esquinas de la ciudad, y M la cantidad de calles **de una mano** entre ellas. Los parámetros `comienzo` y `fin` indican las esquinas de partida y finalización del recorrido. Las esquinas están numeradas desde 1 hasta N , inclusive.

Para cada i entre 0 y $M - 1$, la calle número i va desde la esquina `origen[i]` hasta la esquina `destino[i]`, y recorrerla insume un tiempo `tiempo[i]`.

K indica la cantidad de caminos a calcular.

La función debe devolver un `ARREGLO[K]` de `ENTEROS LARGOS`, con los tiempos correspondientes a los K mejores caminos existentes, ordenados de menor a mayor.

Se garantiza que siempre existirán por lo menos K caminos desde el *comienzo* hasta el *fin*.

Evaluador local

El evaluador local lee la entrada de `stdin` con el siguiente formato:

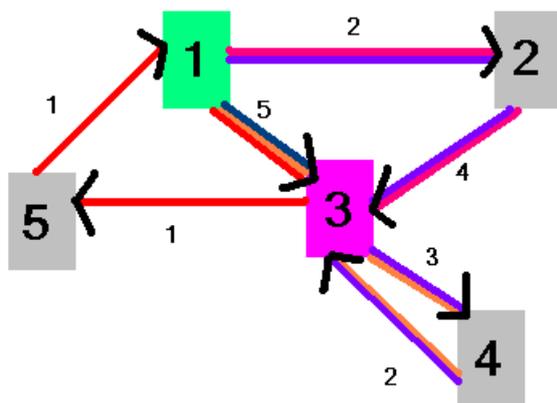
- Una línea con dos enteros N y M
- Una línea con dos enteros `comienzo` y `fin`
- Una línea con un entero K
- M líneas, cada una con tres enteros correspondientes a `origen[i]`, `destino[i]` y `tiempo[i]`

El evaluador ejecutará la función `gps` con estos datos, y mostrará por pantalla la lista de valores que devuelva.

Ejemplo

Si se ejecuta el evaluador con los siguientes datos de entrada:

5	7	
1	3	
5		
1	2	2
1	3	5
2	3	4
5	1	1
3	5	1
3	4	3
4	3	2



Mejores caminos:

- 1) 1-3
- 2) 1-2-3
- 3) 1-3-4-3
- 4) 1-2-3-4-3
- 5) 1-3-5-1-3

En el caso de utilizarlo con un programa correcto, se mostrarán por pantalla los siguientes valores:

5	6	10	11	12
---	---	----	----	----

Que se corresponden con tiempos de los 5 mejores caminos indicados en la imagen.

Cotas

- $1 \leq K \leq 1.000$
- $1 \leq N \leq 1.000$
- $0 \leq M \leq 10.000$
- $0 \leq tiempo[i] \leq 1.000.000$

Subtareas

Habrán casos de prueba por un total de 15 puntos en los cuales $K = 1$.

Habrán casos de prueba por un total de 30 puntos en los cuales $K = 2$.