

## Acortando el camino

*Contribución de Hugo Ryckeboer*

### Descripción del problema

En un cordón montañoso que separa dos valles, arqueólogos localizaron otra ciudad subterránea, al estilo de Petra en Turquía. Posee una salida hacia cada valle.

Diversos exploradores fueron recorriendo las galerías numerando los puntos de bifurcación cada vez que se encontraron con uno nuevo. Al mismo tiempo numeraron las galerías anotando los puntos de bifurcación que unían y sus largos.

Algunas galerías tenían obstruido el paso por derrumbes ocurridos a lo largo de los años. Ya sea por sonido ya sea por mediciones topográficas pudieron identificar desde ambos lados de las obstrucciones los puntos de bifurcación que originariamente unían y calcular sus largos.

En estos momentos es factible con un recorrido largo pasar de un valle a otro usando las galerías de la ciudad subterránea. Aunque largo, el esfuerzo es menor que cruzar por arriba del cordón montañoso.

Disponen de dos cuadrillas con sus respectivos equipos de perforación como para poder remover dos obstrucciones. La finalidad es acortar lo más posible el camino que cruza la montaña usando las galerías.

Como no saben cuales remover piden como ayuda que les proporciones un programa `ciudad.c`, `ciudad.cpp` o `ciudad.pas` que logre identificar las galerías a reabrir.

No tienen certeza de que este trabajo sea útil de modo tal que esperan les proporciones uno de tres resultados:

1. No hay ningún par de galerías obstruidas cuya apertura acorte la distancia actual, dando el valor de esta.
2. No se gana nada abriendo dos, se consigue acortarla con sólo abrir una, proponerla y aportar la distancia resultante de una elección óptima de la misma.

3. Se propone un par que maximiza la mejora y la distancia así resultante.

### Datos de entrada

Se recibe un archivo `ciudad.in` con el siguiente formato:

- Una línea conteniendo tres números indicando la cantidad **B** ( $2 \leq B \leq 10.000$ ) de bifurcaciones (este número incluye la entrada y la salida), la cantidad **G** de galerías abiertas exploradas en la ciudad subterránea ( $1 \leq G \leq 60.000$ ) y la cantidad **O** ( $2 \leq O \leq 100$ ) de galerías obstruidas.
- **G** líneas para describir cada galería. Cada una con tres números: los puntos que une (cada uno entre 1 y **B**) y su largo **l**, ( $1 \leq l \leq 10.000$ )
- **O** líneas con el mismo formato que las anteriores.

**Nota:** Una de las entradas se rotuló **1** y la otra **B**.

### Datos de salida

Se debe generar un archivo `ciudad.out` conteniendo una única línea conteniendo:

- El caso **1**, **2** o **3** que identificaron
- 0, 1 o 2 números identificando galerías obstruidas, cada uno estará en el rango [**1**..**O**]. De ser 2 las galerías a abrir, imprimirlas en el orden en que se las utilizará al ir de **1** a **B**.
- El largo del menor camino que une los valles.
- De haber más de una solución cualquiera de ellas vale.

Ejemplos

Si la entrada `ciudad.in` fuera:

5	6	4
1	2	3
2	4	11
1	3	8
2	3	4
3	4	5
4	5	4
2	5	14
2	3	5
2	3	3
3	5	7

La salida `ciudad.out` debería ser:

3	3	4	13
---	---	---	----

Sin la última línea de `ciudad.in` y `O` valiendo 3 la salida `ciudad.out` debería ser

2	3	15
---	---	----

Quitando otra línea más y `O` valiendo 2 la salida `ciudad.out` debería ser

1	16
---	----

