



International Olympiad in Informatics 2013

6-13 July 2013
Brisbane, Australia

dreaming

Spanish — 1.0

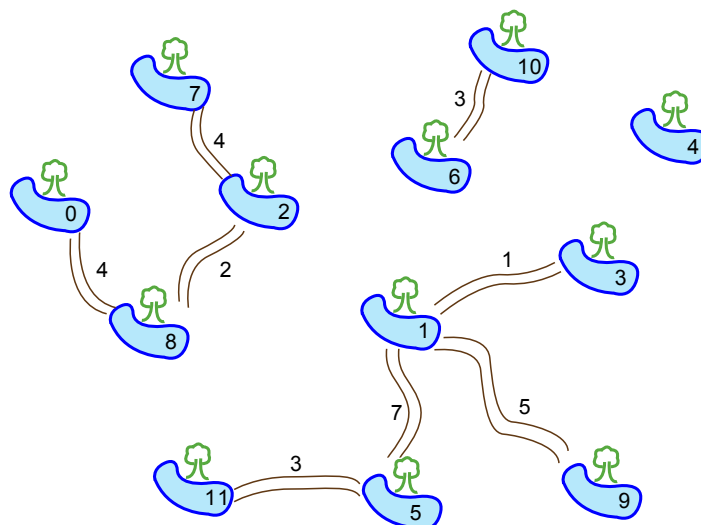
Esta historia toma lugar hace mucho tiempo, cuando el mundo era nuevo y la IOI no había sido soñada.

Serpiente vive en una tierra que tiene N billabongs (agujeros de agua), numerados $0, \dots, N-1$. Hay M caminos bidireccionales uniendo pares de billabongs, por donde Serpiente puede viajar. Cada par de billabongs está conectado (directa o indirectamente) por a lo sumo una secuencia de caminos, pero algunos pares de billabongs puede que no estén conectados (así, $M \leq N-1$). Cada camino le toma un cierto número de días para recorrerlo a Serpiente: este número puede ser diferente para cada camino.

Canguro, amigo de Serpiente, quiere hacer $N - M - 1$ caminos nuevos, para que se le haga posible a Serpiente viajar entre cualquier par de billabongs. Canguro puede crear caminos entre cualquier par de billabongs, y cada camino que Canguro pueda crear le tomará L días a Serpiente para recorrerlos.

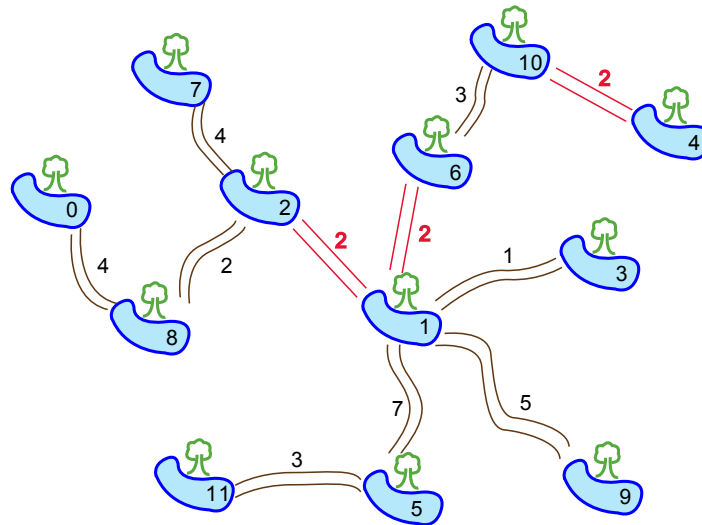
Adicionalmente, Canguro quiere hacer los viajes de Serpiente tan rápidos como sean posibles. Canguro construirá nuevos caminos para que el tiempo de viaje más largo entre cualquier par de billabongs sea lo más corto posible. Ayuda a Canguro y Serpiente a determinar el tiempo de viaje más largo entre cualquier par de billabongs, luego de que Canguro construya los nuevos caminos.

Ejemplos



En la anterior ilustración hay $N = 12$ billabongs y $M = 8$ caminos. Supón que $L = 2$, así cada nuevo camino requerirá 2 días para que Serpiente lo recorra. Entonces Canguro puede construir tres nuevos caminos:

- Entre los billabongs 1 y 2;
- Entre los billabongs 1 y 6;
- Entre los billabongs 4 y 10;



La ilustración anterior muestra el conjunto final de caminos. El viaje más largo es de 18 días, entre los billabongs 0 y 11. Este es el resultado más pequeño posible--sin importar como Canguro construya los caminos, habrá un par de billabongs que requerirán a Serpiente viajar por 18 días o más.

Implementación

Debes enviar un archivo que implemente la función `travelTime()`, como sigue:

Tu Función: `travelTime()`

C/C++

```
int travelTime(int N, int M, int L,
               int A[], int B[], int T[]);
```

Pascal

```
function travelTime(N, M, L : LongInt;
                   var A, B, T : array of LongInt) : LongInt;
```

Descripción

Esta función debe calcular el tiempo de viaje más largo (medido en días) entre cualquier par de billabongs, asumiendo que Canguro agregó $N - M - 1$ caminos de una forma que todos los billabongs están conectados y este tiempo más largo es lo más corto posible.

Parámetros

- N : El número de billabongs
- M : El número de caminos que ya existen.
- L : El tiempo en días que le toma a Serpiente viajar en un nuevo camino.
- A , B y T : Arreglos de longitud M que especifican los puntos finales y el tiempo de viaje de cada camino ya existente, así el i ésimo camino une los billabongs $A[i-1]$ y $B[i-1]$, y toma $T[i-1]$ días en atravesarlo en cualquier dirección.
- *Retorna*: El tiempo máximo entre cualquier par de billabongs, como descrito anteriormente.

Sesión de Ejemplo

La siguiente sesión describe el ejemplo anterior:

Parameter	Value
N	12
M	8
L	2
A	[0, 8, 2, 5, 5, 1, 1, 10]
B	[8, 2, 7, 11, 1, 3, 9, 6]
T	[4, 2, 4, 3, 7, 1, 5, 3]
Returns	18

Restricciones

- Tiempo límite: 1 segundo
- Límite de memoria: 64 MiB (Megabytes)
- $1 \leq N \leq 100,000$
- $0 \leq M \leq N - 1$
- $0 \leq A[i], B[i] \leq N - 1$
- $1 \leq T[i] \leq 10,000$
- $1 \leq L \leq 10,000$

Subtareas

Subtarea	Puntos	Restricciones adicionales en la entrada
1	14	$M = N - 2$, y hay precisamente uno o dos caminos ya existentes que van desde cada billabong. En otras palabras, hay dos conjuntos de billabongs conectados, y en cada conjunto de caminos forman un camino no derivado.
2	10	$M = N - 2$ and $N \leq 100$
3	23	$M = N - 2$
4	18	Hay al menos un camino pre-existente que va desde cada billabong
5	12	$N \leq 3,000$
6	23	(Ninguna)

Experimentación

El evaluador de ejemplo en tu computadora leerá la entrada del archivo `dreaming.in`, que deberá tener el siguiente formato:

- línea 1: `N M L`
- líneas 2, ..., $M + 1$: `A[i] B[i] T[i]`

En instancia, el ejemplo anterior debería ser presentado en el siguiente formato:

```
12 8 2
0 8 4
8 2 2
2 7 4
5 11 3
5 1 7
1 3 1
1 9 5
10 6 3
```

Notas del lenguaje

C/C++ Tu debes `#include "dreaming.h"`.

Pascal Tu debes definir la `unit Dreaming`. Todos los arreglos están numerados comenzando en `0` (no `1`).

Ve la plantillas de las soluciones que se encuentran en tu maquina para ejemplos.