



International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

soñando

Español-Colombia —

1.0

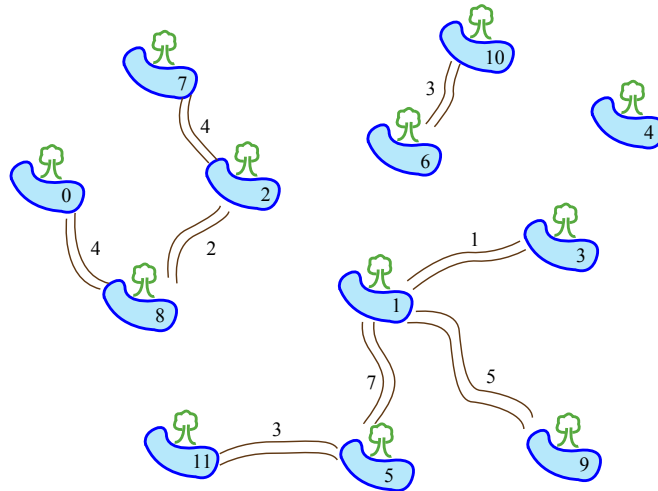
Esta historia ocurre muchos años atrás, la tierra todavía estaba caliente y la IOI ni siquiera era un sueño.

Serpiente vive en un mundo con N billabongs (estanques de agua), numerados $0, \dots, N - 1$. Serpiente puede recorrer M caminos bidireccionales que unen billabongs. Cada par de billabongs está conectado (directa o indirectamente) con a lo sumo una única secuencia de caminos. Sin embargo, pueden existir pares de billabongs no conectados entre sí (por ende, $M \leq N - 1$). Serpiente se demora una determinada cantidad de días en recorrer cada camino, este número puede variar entre caminos.

El amigo de Serpiente, Canguro, quiere hacer $N - M - 1$ nuevos caminos de tal forma que Serpiente pueda ir por alguna secuencia de caminos desde un billabong a cualquier otro. Canguro puede crear caminos nuevos entre cualquier par de billabongs. Serpiente se demora L días en recorrer alguno de estos nuevos caminos.

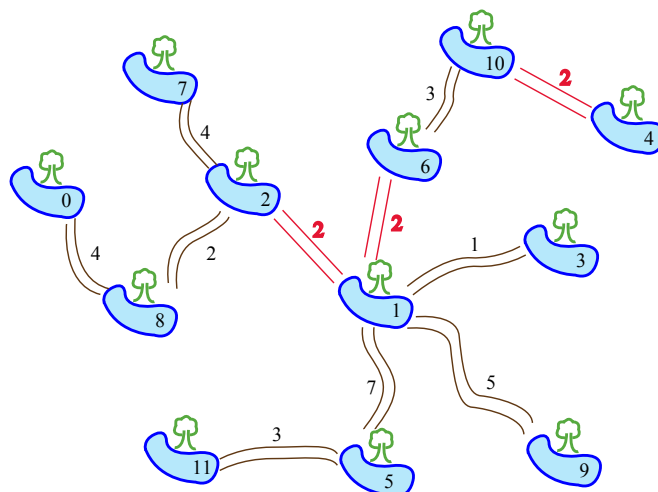
Además, Canguro quiere asegurarse que los viajes de Serpiente sean lo más corto posible. Para esto, Canguro construirá nuevos caminos de tal forma que la longitud del viaje más largo entre cualquier par de billabongs sea lo menor posible. Ayude a Canguro y a Serpiente a determinar la longitud del viaje más largo entre algún par de billabongs, luego de que Canguro haya construido nuevos caminos.

Ejemplos



La figura anterior muestra $N = 12$ billabongs y $M = 8$ caminos. Suponga que $L = 2$, por lo cual Serpiente demorará dos días en recorrer cualquier camino que Canguro cree. Por lo tanto, Canguro creará los siguientes tres nuevos caminos:

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



La figura anterior muestra los caminos construidos por Canguro. El viaje más largo, entre los billabongs 0 y 11, demora 18 días. Este es el resultado más corto posible, no importa cómo construya Canguro los caminos, siempre habrá un viaje entre algún par de billabongs que demore 18 o más días.

Implementación

Envíe un archivo implementando la función `travelTime()`, de la siguiente forma:

Su 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

La función debe calcular la longitud del viaje más largo (medido en días) entre cualquier par de billabongs, asumiendo que Canguro ya añadió $N - M - 1$ caminos de tal forma que todos los billabongs están conectados y la longitud del viaje más largo sea lo más corto posible.

Parámetros

- `N`: El número de billabongs.
- `M`: El número de caminos que ya existen.
- `L`: El tiempo en días que se demora Serpiente en recorrer cualquier camino nuevo.
- `A`, `B` y `T`: Arreglos de longitud `M` que delimitan los extremos y el tiempo de viaje de cada camino existente. El camino número `i` une los billabongs `A[i-1]` y `B[i-1]`, y Serpiente se demora `T[i-1]` días recorriendo ese camino en cualquier dirección.
- *Returns*: La longitud del camino más largo descrito anteriormente.

Sesión de Prueba

La siguiente sesión muestra el ejemplo descrito anteriormente:

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

- Límite de tiempo: 1 segundo
- Límite de memoria: 64 MB
- $1 \leq N \leq 100,000$
- $0 \leq M \leq N - 1$
- $0 \leq A[i], B[i] \leq N - 1$
- $1 \leq T[i] \leq 10,000$
- $1 \leq L \leq 10,000$

Subtareas

Subtarea	Puntos	Restricciones adicionales
1	14	$M = N - 2$, y existen exactamente uno o dos caminos pre-existentes desde cada billabong. Es decir, existen dos conjuntos de billabongs conectados entre sí, y cada conjunto es un solo camino sin ramificaciones.
2	10	$M = N - 2$ y $N \leq 100$
3	23	$M = N - 2$
4	18	Existe por lo menos un camino pre-existente desde cada billabong.
5	12	$N \leq 3,000$
6	23	(Ninguna)

Experimentación

El calificador en su máquina lee la entrada del archivo `dreaming.in`, que debe estar en el siguiente formato:

- Línea 1: `N M L`
- Líneas 2, ..., `M + 1`: `A[i] B[i] T[i]`

Por ejemplo, lo mostrado anteriormente se redactaría 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 de Lenguaje

C/C++ You must `#include "dreaming.h"`.

Pascal You must define the `unit Dreaming`. All arrays are numbered beginning at `0` (not `1`).

See the solution templates on your machine for examples.