



**International Olympiad in Informatics**  
**2013**  
 6-13 July 2013  
 Brisbane, Australia

**dreaming**  
 Spanish — 1.0

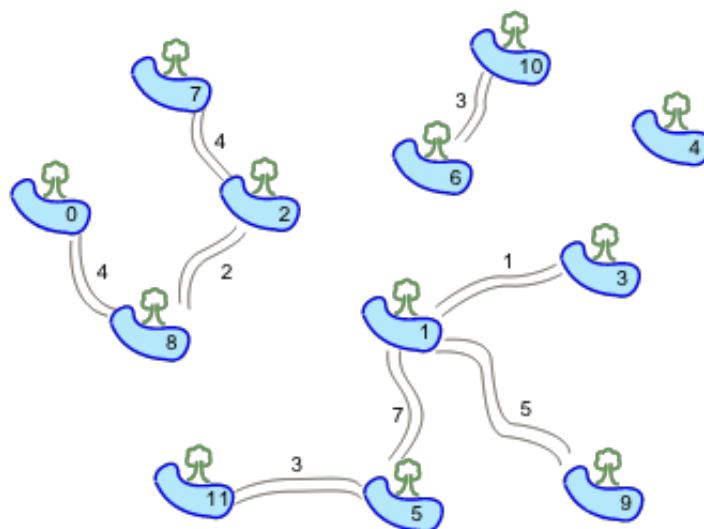
Esta historia tiene lugar mucho tiempo atrás, cuando el mundo era nuevo y la IOI ni siquiera se había soñado.

Una Serpiente vive en una tierra formada por  $N$  billabongs (hoyos de agua) numerados de la forma  $0, \dots, N-1$ . Existen  $M$  caminos bidireccionales que unen pares de billabongs, por los cuales la serpiente puede viajar, cada par de billabongs se encuentra conectado (directa o indirectamente) por a lo más una secuencia de caminos, y puede que algunos pares de billabongs no estén conectados (es decir,  $M \leq N-1$ ). Viajar por cada camino le toma a la Serpiente cierto número de días, y este número puede ser diferente para cada camino.

Un Canguro, amigo de la serpiente, quiere hacer  $N - M - 1$  nuevos caminos, de tal forma que le sea posible a la Serpiente viajar entre cualquier par de billabongs. El Canguro puede crear caminos entre cualquier par de billabongs, y cada camino creado por el Canguro le tomará  $L$  días a la Serpiente viajar a través de él.

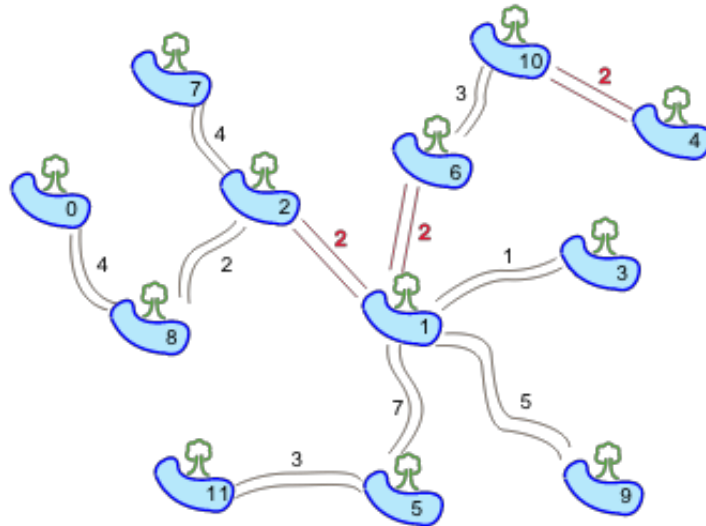
Adicionalmente, el Canguro busca hacer los viajes de la Serpiente lo más rápidos que sea posible, así que construirá los caminos de forma que el viaje más largo entre cualesquiera dos billabongs sea lo más corto posible. Ayuda al Canguro y a la serpiente a determinar el viaje más largo entre dos billabongs después de que el Canguro haya construido los nuevos caminos.

## Ejemplos



En la figura de arriba existen  $N = 12$  billabongs y  $M = 8$  caminos. Supón que  $L = 2$ , es decir que cada nuevo camino le tomará a la Serpiente 2 días de viaje. Entonces el Canguro podría construir tres nuevos caminos:

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



La figura de arriba muestra el conjunto final de caminos. El viaje más largo es de 18 días, y es entre el billabong 0 y el 11. Este es el resultado más corto posible, es decir, no importa como el Canguro construya los caminos, existirá algún par de billabongs para el cual la Serpiente necesite viajar por 18 días o más.

## Implementación

Debes enviar un archivo implementando la función `travelTime()`, como se describe a continuación:

### 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 más largo de viaje (medido en días) entre cualquier par de billabongs, asumiendo que el Canguro añadió  $N - M - 1$  caminos de modo que todos los billabongs estén conectados y este tiempo más largo de viaje sea lo más pequeño posible.

## Parámetros

- $N$ : El número de billabongs.
- $M$ : El número de caminos que existen.
- $L$ : El tiempo en días que le tomará a la Serpiente viajar a través de cada nuevo camino.
- $A$ ,  $B$  and  $T$ : Arreglos de longitud  $M$  que especifican los puntos finales y el tiempo de viaje para cada camino pre existente, de modo que el  $i$ -ésimo camino une los billabongs  $A[i-1]$  y  $B[i-1]$ , y toma  $T[i-1]$  días viajar en cualquiera de las dos direcciones.
- *Returns*: El tiempo de viaje más largo entre cualquier par de billabongs como se describe arriba.

## Sesión Ejemplo

La siguiente sesión describe el ejemplo de arriba:

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

## Restricciones

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

## Subtareas

Subtarea	Puntos	Restricciones Adicionales a la Entrada
1	14	$M = N - 2$ , y hay precisamente uno o dos caminos que llegan a cada billabong, en otras palabras, existen dos conjuntos de billabongs conectados, y en cada conjunto los caminos forman una trayectoria sin ramificaciones.
2	10	$M = N - 2$ y $N \leq 100$
3	23	$M = N - 2$
4	18	Para cada billabong no hay más de un camino que termine en él.
5	12	$N \leq 3,000$
6	23	(Ninguna)

## Experimentación

El evaluador de ejemplo en tu computadora leerá la entrada del archivo `dreaming.in`, el cual debe seguir el siguiente formato:

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

Por ejemplo, el ejemplo de arriba debiera ser provisto 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++ Deberás incluir `#include "dreaming.h"`.

Pascal Deberás definir la Unidad `unit Dreaming`. Todos los arreglos están numerados iniciando en `0` (no `1`).

Revisa los templates de las soluciones dentro de tu computadora para ver ejemplo.

