



## International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

# dreaming

Norwegian — 1.0

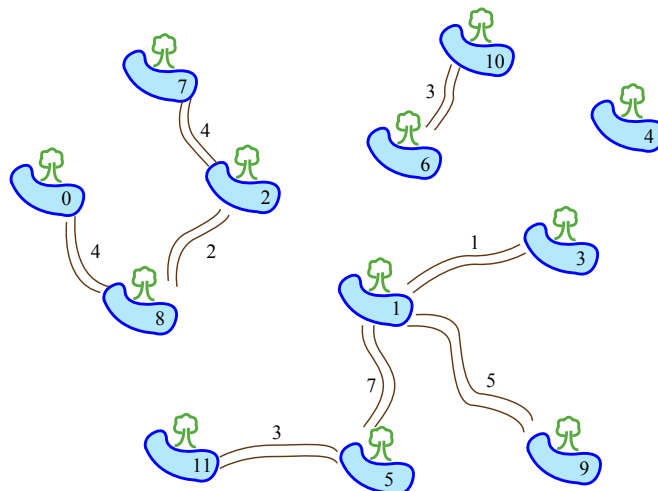
Denne historien finner sted for lenge siden, da verden var ny og IOI ikke ennå hadde blitt drømt.

Slange lever i et land som har  $N$  billabonger (vannhull), som er nummerert  $0, \dots, N - 1$ . Det finnes  $M$  toveis stier som kobler sammen par av billabonger, som Slange kan bevege seg langs. Hvert par av billabonger er koblet sammen (direkte eller indirekte) av maksimalt én sekvens av stier, skjønt det kan hende at noen par av billabonger ikke er koblet sammen i det hele tatt (så  $M \leq N - 1$ ). Slange trenger et visst antall dager for å bevege seg langs hver sti; dette antallet kan variere fra sti til sti.

Slanges venn, Kenguru, ønsker å lage  $N - M - 1$  nye stier, slik at det er mulig for Slange å bevege seg mellom et hvilket som helst par av billabonger. Kenguru kan lage stier mellom et hvilke som helst par av billabonger, og hver sti som Kenguru lager vil kreve  $L$  dager for Slange å bevege seg langs.

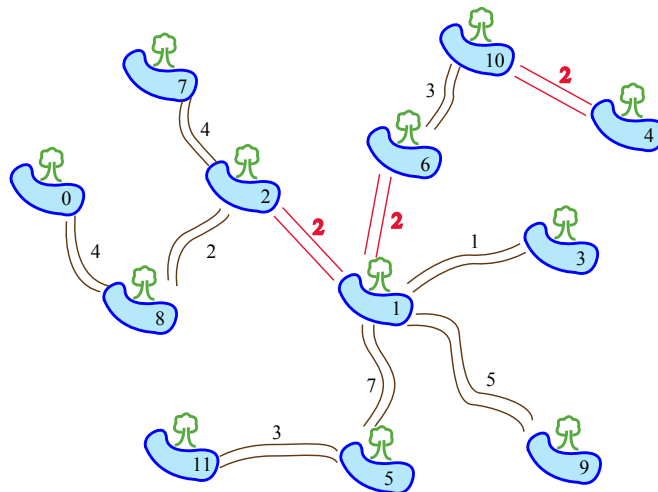
I tillegg vil Kenguru gjøre Slanges reiser så raske som mulig. Kenguru vil bygge de nye stiene slik at den lengste reisetiden mellom hvilket som helst par av billabonger er så liten som mulig. Hjelp Kenguru og Slange med å avgjøre den lengste reisetiden mellom hvilket som helst par av billabonger etter at Kenguru har bygd de nye stiene på denne måten.

## Examples



I bildet ovenfor er det  $N = 12$  billabonger og  $M = 8$  stier. Anta at  $L = 2$ , slik at hver nye sti krever 2 dager for at Slange skal krysse den. Da kan Kenguru bygge tre nye stier:

- Mellom billabongene 1 og 2
- Mellom billabongene 1 og 6
- Mellom billabongene 4 og 10



Bildet ovenfor viser den endelige samlingen med stier. Den lengste reisetiden er 18 dager, mellom vannhull 0 og 11. Dette er det minste mulige resultatet - uansett hvordan Kenguru bygger stiene vil det finnes et par av vannhull som Slange trenger 18 dager eller mer for å reise mellom.

---

## Implementation

You should submit a file implementing the function `travelTime()`, as follows:

**Your Function:** `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;
```

### Description

This function should calculate the greatest travel time (measured in days) between any pair of billabongs, assuming that Kangaroo has added  $N - M - 1$  trails in such a way that all billabongs are connected and this greatest travel time is as small as possible.

### Parameters

- `N`: The number of billabongs.
- `M`: The number of trails that already exist.
- `L`: The time in days that it takes Serpent to travel along a new trail.
- `A`, `B` and `T`: Arrays of length `M` that specify the endpoints and travel time of each pre-existing trail, so that the  $i$ th trail joins billabongs `A[i-1]` and `B[i-1]`, and takes `T[i-1]` days to travel in either direction.
- *Returns*: The greatest travel time between any pair of billabongs, as described above.

---

## Sample Session

The following session describes the example above:

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

---

## Constraints

- Time limit: 1 second
- Memory limit: 64 MiB
- $1 \leq N \leq 100,000$
- $0 \leq M \leq N - 1$
- $0 \leq A[i], B[i] \leq N - 1$
- $1 \leq T[i] \leq 10,000$
- $1 \leq L \leq 10,000$

---

## Subtasks

Subtask	Points	Additional Input Constraints
1	14	$M = N - 2$ , and there are precisely one or two pre-existing trails leading from each billabong. In other words, there are two sets of connected billabongs, and in each set the trails form an unbranching path.
2	10	$M = N - 2$ and $N \leq 100$
3	23	$M = N - 2$
4	18	There is at most one pre-existing trail leading from each billabong.
5	12	$N \leq 3,000$
6	23	(None)

---

## Experimentation

The sample grader on your computer will read input from the file `dreaming.in`, which must be in the following format:

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

For instance, the example above should be provided in the following format:

```
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
```

---

## Language Notes

**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.