



## International Olympiad in Informatics

2013

6-13 July 2013

Brisbane, Australia

# dreaming

Netherlands — 1.0

Lang lang geleden, toen de wereld nog nieuw was en de IOI nog niet bedacht was...

Oerslang leeft in een land met  $N$  billabongs (meren) die als volgt genummerd zijn:

0, ...,  $N - 1$  .

Er zijn  $M$  paden die in in twee richtingen gevolgd kunnen worden door Oerslang. Elk pad verbindt een tweetal billabongs.

Er is hoogstens één reeks van paden die een willekeurig paar billabongs (direct of indirect) verbindt. Maar niet elk paar billabongs hoeft verbonden te zijn (dus,  $M \leq N - 1$ ).

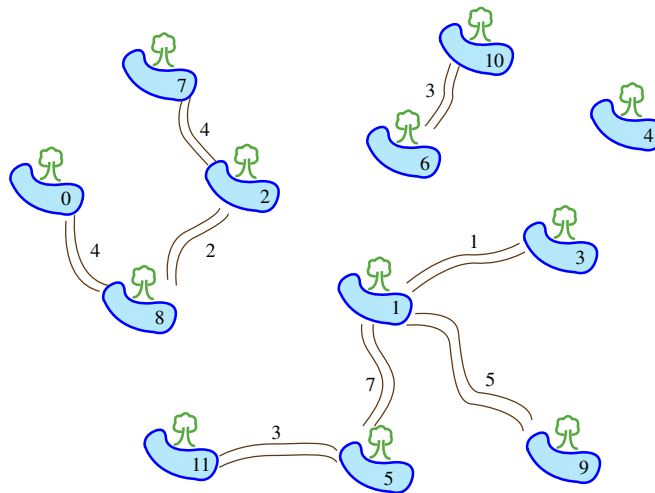
Voor elk pad kost het Oerslang een bepaald aantal dagen om over het pad te reizen. Dit aantal dagen kan voor elk pad anders zijn.

Oerslang heeft een vriendje, Kangoeroe, die  $N - M - 1$  nieuwe paden wil maken, zodat het voor Oerslang mogelijk wordt om tussen elk paar billabongs te reizen. Kangoeroe kan een pad maken tussen elk paar billabongs. Het kost Oerslang  $L$  dagen om over een nieuw aangelegd pad te reizen.

Kangoeroe wil graag dat Oerslang zo snel mogelijk kan reizen. Hij wil de nieuwe paden zo aanleggen dat de langste reistijd tussen een willekeurig paar billabongs zo kort mogelijk is. Help Kangoeroe en Oerslang te bepalen wat de langste van alle reistijden tussen een tweetal billabongs is, nadat Kangoeroe de nieuwe paden heeft aangelegd.

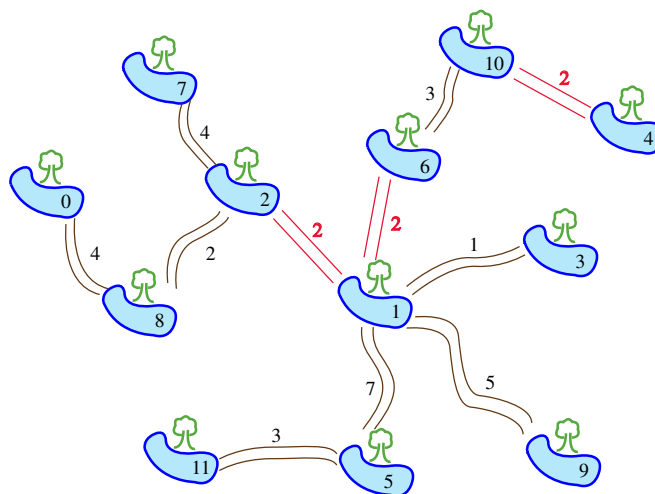
---

## Voorbeelden



In het plaatje hierboven staan  $N=12$  billabongs en  $M=8$  paden. Stel dat  $L=2$ , zodat Oerslang voor elk nieuw aangelegd pad 2 dagen nodig heeft om dit te bereizen. Kangoeroe legt de volgende drie nieuwe paden aan:

- tussen billabongs 1 en 2;
- tussen billabongs 1 en 6;
- tussen billabongs 4 en 10.



In het plaatje hierboven zie je alle bestaande paden als Kangoeroe klaar is. De langste reistijd is 18 dagen, tussen de billabongs 0 en 11. Dit is het kleinst mogelijke resultaat - hoe Kangoeroe de paden ook aanlegt, er zal altijd een paar billabongs zijn waarbij Oerslang er 18 of meer dagen over doet om tussen deze billabongs te reizen.

---

## Implementatie

Je moet een bestand insturen dat de functie `travelTime()` als volgt implementeert:

**Jouw functie:** `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;

```

## Beschrijving

Deze functie moet de grootste van alle reistijden (gemeten in dagen) tussen een paar billabongs bepalen. Ga er hierbij vanuit dat Kangoeroe  $N - M - 1$  paden aanlegt zodat alle billabongs verbonden zijn, en de grootste reistijd zo klein mogelijk is.

## Parameters

- $N$  : Het aantal billabongs.
- $M$  : Het aantal reeds bestaande paden.
- $L$  : De reistijd (in dagen) voor Oerslang over een nieuw aangelegd pad.
- $A$ ,  $B$  en  $T$  : Arrays van lengte  $M$  die de eindpunten en reistijd aangeven van elk pad dat reeds bestaat. Het  $i$ -de pad verbindt de billabongs  $A[i-1]$  en  $B[i-1]$ , en heeft een reistijd van  $T[i-1]$  dagen in beide richtingen.
- *Resultaat*: De hoogste van alle reistijden tussen een tweetal billabongs zoals hierboven beschreven.

---

## Voorbeeld

Hieronder wordt het voorbeeld van hierboven uitgewerkt:

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

---

## Randvoorwaarden

- Tijdslimiet: 1 seconde
- Geheugenlimiet: 64 Mb

- $1 \leq N \leq 100.000$
- $0 \leq M \leq N - 1$
- $0 \leq A[i], B[i] \leq N - 1$
- $1 \leq T[i] \leq 10.000$
- $1 \leq L \leq 10.000$

## Subtasks

Subtask	Punten	Aanvullende randvoorwaarden
1	14	$M = N - 2$ , en er zijn precies een of twee bestaande paden die leiden vanuit elke billabong. Met andere woorden: er zijn twee sets van verbonden billabongs, en in elke set vormen de paden een wandeling zonder vertakkingen.
2	10	$M = N - 2$ en $N \leq 100$
3	23	$M = N - 2$
4	18	Er is maximaal een reeds bestaand pad vanuit elke billabong.
5	12	$N \leq 3.000$
6	23	(Geen)

## Stoeien

De voorbeeld grader op je computer leest de invoer uit het bestand `dreaming.in`, dat in het volgende formaat is:

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

Het voorbeeld hierboven wordt in het volgende formaat aangegeven:

```
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
```

## Opmerkingen bij de talen

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.