



**International Olympiad in
Informatics 2013**

6-13 July 2013
Brisbane, Australia

dreaming

Nederlands (België) — 1.0

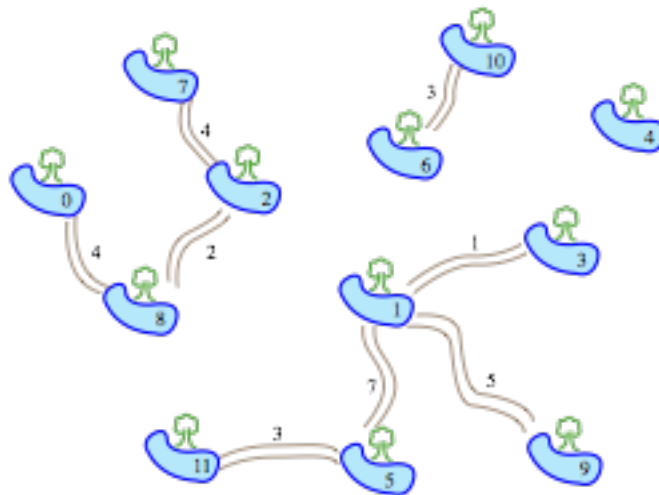
Dit verhaal gebeurde lang geleden, toen de wereld nieuw was en de IOI nog niet was gedroomd.

Slang leeft in een land met N billabongs (waterputten), genummerd $0, \dots, N-1$. Er zijn M tweerichtings-*paden*, die paren van billabongs verbinden, en waarlangs Slang kan bewegen. Elk paar van billabongs is verbonden, direct of indirect, door maximaal één reeks paden. Er kunnen echter wel paren van billabongs zijn die niet met elkaar verbonden zijn (dus: $M \leq N-1$). Om langs een pad te reizen heeft Slang een zeker aantal dagen nodig: dit aantal kan verschillend zijn voor elk pad.

Kangoeroe, de vriend van Slang, wil $N - M - 1$ nieuwe paden maken, zodat Slang kan reizen tussen eender welk paar billabongs. Kangoeroe kan paden maken tussen eender welk paar billabongs, en Slang heeft L dagen nodig om te reizen langs een pad dat Kangoeroe heeft gemaakt.

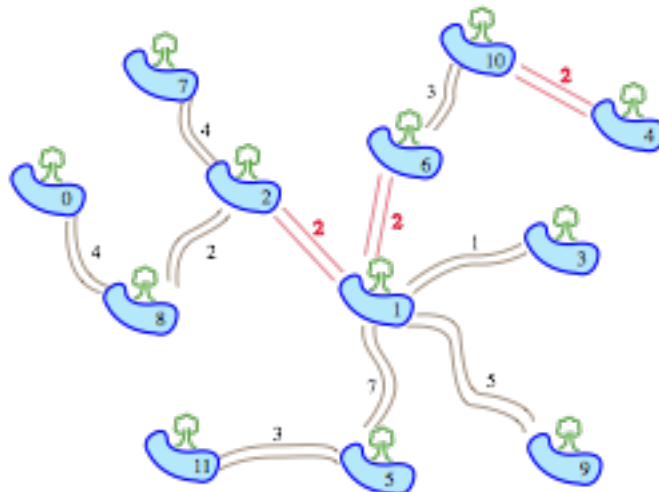
Bovendien wil Kangoeroe dat Slang zo snel mogelijk kan reizen. Kangoeroe zal de nieuwe paden bouwen, zodat de langste reistijd tussen elk paar billabongs zo kort mogelijk is. Help Kangoeroe en Slang met het bepalen van de maximale reistijd tussen twee billabongs, nadat Kangoeroe de nieuwe paden op deze manier heeft gebouwd.

Voorbeelden



In de figuur hierboven zijn er $N = 12$ billabongs en $M = 8$ paden. Stel dat $L = 2$, zodat Slang 2 dagen nodig heeft om langs een nieuw pad te reizen. Dan kan Kangoeroe drie nieuwe paden bouwen:

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



De figuur hierboven toont de uiteindelijke verzameling paden. De langste reistijd is 18 dagen, tussen billabongs 0 en 11. Dit is het kleinst mogelijke resultaat - het maakt niet uit hoe Kangoeroe de paden bouwt, er zal steeds een paar billabongs zijn waartussen Slang 18 of meer dagen zal moeten reizen.

Implementatie

Je moet een bestand indienen 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 langste reistijd (gemeten in dagen) tussen elk paar billabongs bepalen. Je mag ervan uitgaan dat Kangoeroe $N - M - 1$ paden heeft aangelegd, zodat alle billabongs verbonden zijn, en zodat de langste reistijd zo klein mogelijk is.

Parameters

- `N` : Het aantal billabongs.
- `M` : Het aantal reeds bestaande paden.
- `L` : De reistijd voor Slang over een nieuw aangelegd pad.
- `A`, `B` en `T` : Arrays van lengte `M` die de eindpunten en reistijd aangeven van elk reeds bestaand pad. Het `i`-de pad verbindt de billabongs `A[i-1]` en `B[i-1]`, en vereist een reistijd van `T[i-1]` dagen in beide richtingen.
- *Returns*: De langste reistijd tussen elk paar billabongs zoals hierboven beschreven.

Voorbeeldscenario

Het volgende scenario beschrijft het voorbeeld hierboven:

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

Beperkingen

- Tijdslimiet: 1 seconde
- Geheugenlimiet: 64 MiB
- $1 \leq N \leq 100,000$
- $0 \leq M \leq N - 1$
- $0 \leq A[i], B[i] \leq N - 1$
- $1 \leq T[i] \leq 10,000$
- $1 \leq L \leq 10,000$

Subtaken

Subtaak	Punten	Aanvullende inputbeperkingen
1	14	$M = N - 2$, en er vertrekken precies één of twee bestaande paden 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 vertrekt maximaal één reeds bestaand pad vanuit elke billabong.
5	12	$N \leq 3,000$
6	23	(Geen)

Experimenteren

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

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

Het voorbeeld hierboven wordt dus in het volgende formaat opgesteld:

```
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
```

Taalspecifieke Opmerkingen

C/C++ Je moet `#include "dreaming.h"` gebruiken.

Pascal Je moet de `unit Dreaming` definiëren. Alle arrays zijn genummerd vanaf `0` (niet `1`).

Bekijk de modeloplossing op je computer als voorbeeld.