



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

**dreaming**

Estonian — 1.0

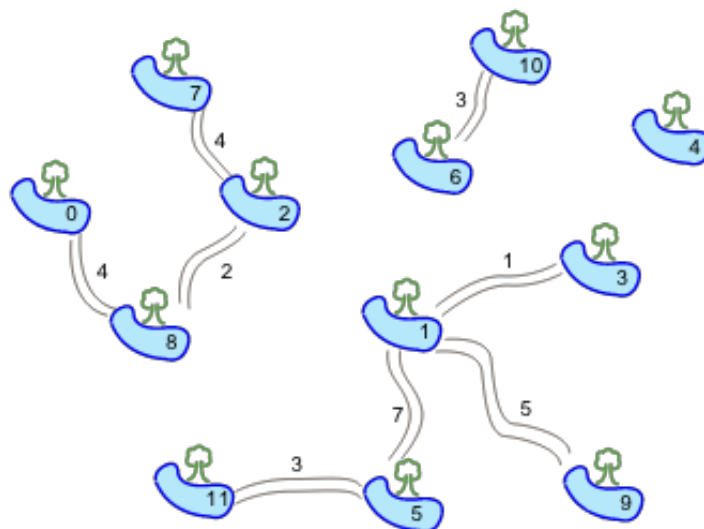
Selle loo tegevus leidis aset ammu-ammu, kui maailm oli alles loodud ja IOI'st polnud veel unistatudki.

Madu elab maal, kus on  $N$  sooti (veesilma), mis on nummerdatud  $0, \dots, N - 1$ . Lisaks on  $M$  rada, millest igaüks ühendab kaht sooti ja mida mööda Madu saab mõlemas suunas liikuda. Iga soodipaar on ühendatud (kas otse või läbi teiste sootide) ülimalt ühe radade jadaga, aga mõnede sootide vahel võib ühendus ka puududa (seega  $M \leq N - 1$ ). Maol kulub iga raja läbimiseks kindel arv päevi (mis võib erinevatel radadel olla erinev).

Mao sõber Känguru tahab luua  $N - M - 1$  uut rada, et maal oleks võimalik liikuda igast soodist igasse teise sooti. Känguru võib uusi radasid teha ükskõik milliste sootide vahele ja Maol kulub iga uue raja läbimiseks  $L$  päeva.

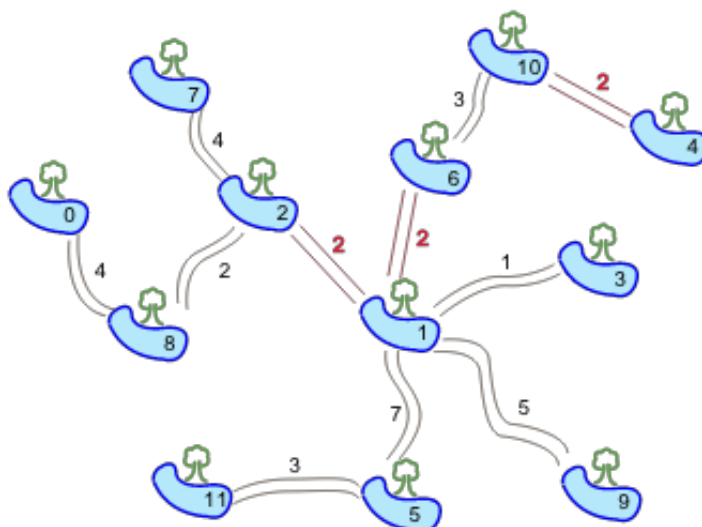
Lisaks tahab Känguru, et Madu saaks reisida võimalikult kiiresti. Känguru loob uued rajad nii, et maksimaalne ühest soodist teise liikumiseks kuluv aeg oleks võimalikult väike. Aita Kängurul ja Maol leida maksimaalne ühest soodist teise liikumiseks kuluv aeg pärast seda, kui Känguru on uued rajad optimaalselt loonud.

## Näited



Ülaloleval joonisel on  $N = 12$  sooti ja  $M = 8$  rada. Olgu  $L = 2$ , ehk Maol kulub iga uue raja läbimiseks 2 päeva. Siis võiks Känguru uued rajad luua järgmiselt:

- sootide 1 ja 2 vahele;
- sootide 1 ja 6 vahele;
- sootide 4 ja 10 vahele.



Ülaloleval joonisel on saadav radade skeem. Maksimaalne reisiks kuluv aeg on 18 päeva (nii palju kulub sootide 0 ja 11 vahel liikumiseks). See on minimaalne võimalik tulemus — ükskõik kuidas Känguru uusi radasid looks, on alati mõni sootide paar, mille vahel liikumiseks kulub 18 päeva või rohkem.

## Realisatsioon

Lahendusena tuleb esitada fail, mis sisaldab funktsiooni `travelTime()`.

### Sinu funktsiooni `travelTime()` deklaratsioon:

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;
```

### Kirjeldus

See funktsioon peab leidma pikima kahe soodi vaheliseks reisiks kuluva aja (päevades) eeldusel, et Känguru on loonud  $N - M - 1$  uut rada nii, et kõik soodid on omavahel ühendatud ja see pikima reisi aeg on võimalikult väike.

### Parameetrid

- $N$  : sootide arv.

- $M$ : olemasolevate radade arv.
- $L$ : Maol uue raja läbimiseks kuluv aeg.
- $A$ ,  $B$  ja  $T$ : massiivid pikkusega  $M$ , mis kirjeldavad olemasolevate radade otspunkte ja nende läbimiseks kuluvat aega; rada number  $i$  ühendab omavahel soote  $A[i-1]$  ja  $B[i-1]$  ning Maol kulub selle läbimiseks ükskõik kummas suunas  $T[i-1]$  päeva.
- *Tagastab*: maksimaalne ühest soodist teise jõudmiseks kuluv aeg, nagu eelpool kirjeldatud.

## Interaktsiooni näide

Järgnev interaktsioon vastab eelkirjeldatud näitele:

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

## Piirangud

- Ajalimit: 1 sekund
- Mälulimit: 64 MiB
- $1 \leq N \leq 100\,000$
- $0 \leq M \leq N - 1$
- $0 \leq A[i], B[i] \leq N - 1$
- $1 \leq T[i] \leq 10\,000$
- $1 \leq L \leq 10\,000$

## Alamülesanded

Alamülesanne	Punkte	Lisapiirengud sisendile
1	14	$M = N - 2$ ja igast soodist väljub kas üks või kaks olemasolevat rada; teisisõnu, soodid on jagatud kahte gruppi ja kummagi grupi rajad moodustavad ühe hargnemisteta jada
2	10	$M = N - 2$ ja $N \leq 100$
3	23	$M = N - 2$
4	18	Igast soodist väljub ülimalt üks olemasolev rada
5	12	$N \leq 3\,000$
6	23	<i>Lisapiiranguid pole</i>

## Katsetamine

Sinu arvutis ülesande materjalide hulgas olev hindamisprogramm loeb sisendi failist `dreaming.in`, mis peab olema järgmises vormingus:

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

Ülalkirjeldatud näide peaks seega olema esitatud kujul:

```
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
```

## Keelespetsiifilised märkused

C/C++ Sa pead kaasama: `#include "dreaming.h"`.

Pascal Sa pead defineerima: `unit Dreaming`. Kõigi massiivide indeksid algavad 0 st (mitte 1 st).

Vaata näidetena ka oma arvutis olevaid programmipõhju.