



International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

dreaming

Finnish — 1.0

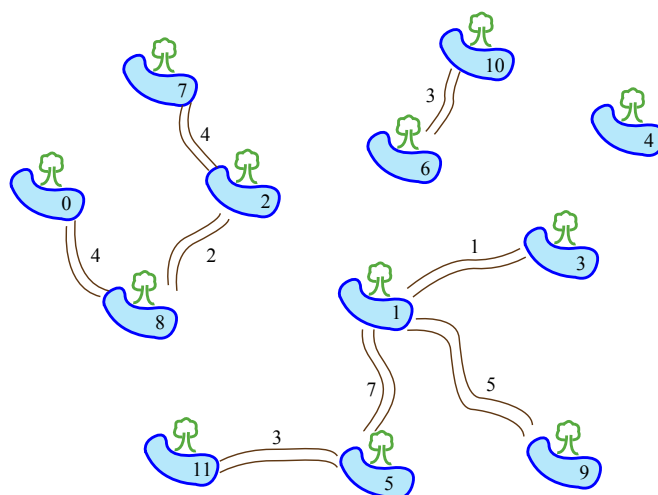
Tämä tarina tapahtuu kauan sitten, kun maailma oli uusi ja IOI:sta ei nähty vielä unta.

Serpent asuu maassa, jossa on N järveä, jotka on numeroitu $0, \dots, N-1$. Lisäksi maassa on M kaksisuuntaista *polkua*, jotka yhdistävät järviä ja joiden kautta Serpent voi kulkea. Jokaiset kaksi järveä on yhdistetty (suoraan tai epäsuoraan) korkeintaan yhden polkuketjun kautta, ja kaikkien järvien välillä ei ole välttämättä yhteyttä (eli $M \leq N-1$). Jokaisen polun kulkemiseen kuluu Serpentiltä tietty määrä päiviä: tämä määrä saattaa olla eri jokaiselle polulle.

Serpentin ystävä, Kangaroo, haluaa luoda $N - M - 1$ uutta polkua, jotta Serpent voisi kulkea minkä tahansa kahden järven välillä. Kangaroo voi luoda polkuja minkä tahansa kahden järven välille, ja jokaisen Kangaroon luoman polun kulkeminen kestää Serpentiltä L päivää.

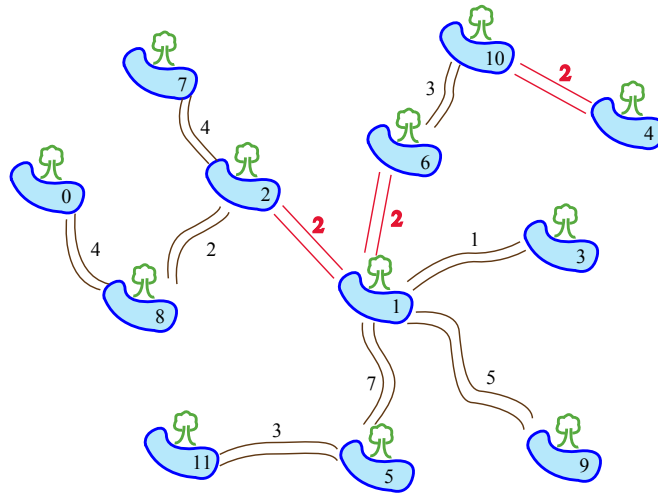
Lisäksi Kangaroo haluaa, että Serpentin matkat ovat mahdollisimman nopeita. Kangaroo luo uudet polut niin, että pisin matkustusaika minkä tahansa kahden järven välillä on mahdollisimman pieni. Auta Kangaroota ja Serpentä määrittämään pisin matkustusaika kahden järven välillä sen jälkeen, kun Kangaroo on luonut uudet polut tällä tavalla.

Esimerkit



Alla olevassa kuvassa on $N = 12$ järveä ja $M = 8$ polkua. Oletetaan, että $L = 2$ eli jokaisen uuden polun kulkemiseen menee Serpentiltä 2 päivää. Tällöin Kangaroo voisi luoda kolme uutta polkua:

- järvien 1 ja 2 välille;
- järvien 1 ja 6 välille;
- järvien 4 ja 10 välille.



Yllä oleva kuva näyttää lopulliset polut. Pisin matkustusaika on 18 päivää, järvien 0 ja 11 välillä. Tämä on pienin mahdollinen tulos - miten tahansa Kangaroo luokki polut, jonkin kahden järven välillä kulkemiseen kuluu 18 päivää tai enemmän.

Toteutus

Sinun tulee lähettää tiedosto, jossa on funktion `travelTime()` toteutus seuraavasti:

Sinun funktiosi: `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;
```

Kuvaus

Tämän funktion tulee laskea suurin matkustusaika (päivinä) kahden järven välillä, kun oletetaan, että Kangaroo on lisännyt $N - M - 1$ polkua sillä tavalla, että kaikki järvet on yhdistetty ja suurin matkustusaika on mahdollisimman pieni.

Parametrit

- **N** : Järvien määrä.
- **M** : Valmiina olevien polkujen määrä.
- **L** : Aika päivinä, joka Serpentiltä kuluu kulkea uusi polku.
- **A** , **B** ja **T** : Taulukot kokoa **M** , jotka kuvaavat kunkin valmiina olevan polun pääte pisteet ja kulkuajan. Kuvauksessa **i** :s polku yhdistää järvet **A[i-1]** ja **B[i-1]** , ja polun kulkemiseen kuluu **T[i-1]** päivää kumpaankin suuntaan.
- *Palauttaa*: Suurin matkustusaika minkä tahansa kahden järven välillä yllä kuvatun mukaisesti.

Esimerkki-istunto

Seuraava istunto vastaa edellä ollutta esimerkkiä:

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

Rajat

- Aikaraja: 1 sekunti
- Muistiraja: 64 MiB
- $1 \leq N \leq 100,000$
- $0 \leq M \leq N - 1$
- $0 \leq A[i], B[i] \leq N - 1$
- $1 \leq T[i] \leq 10,000$
- $1 \leq L \leq 10,000$

Alitehtävät

Alitehtävä	Pisteet	Syötteen lisärajoitukset
1	14	M = N - 2 ja jokaisesta järvestä lähtee aloitustilanteessa tarkalleen yksi tai kaksi polkua. Toisin sanoen järvet muodostavat kaksi yhdistettyä joukkoa, ja kummassakin joukossa poluista muodostuu reitti, jossa ei ole haarautumia.
2	10	$M = N - 2$ ja $N \leq 100$
3	23	$M = N - 2$
4	18	Jokaisesta järvestä lähtee korkeintaan yksi polku aloitustilanteessa.
5	12	$N \leq 3,000$
6	23	(Ei mitään)

Kokeilu

Esimerkkiarvostelija lukee syötteen tiedostosta `dreaming.in`, jonka muodon täytyy olla seuraava:

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

Esimerkiksi yllä oleva esimerkki tulisi antaa seuraavassa muodossa:

```
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
```

Huomioita kielistä

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.