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

dreaming

Srpski — 1.0

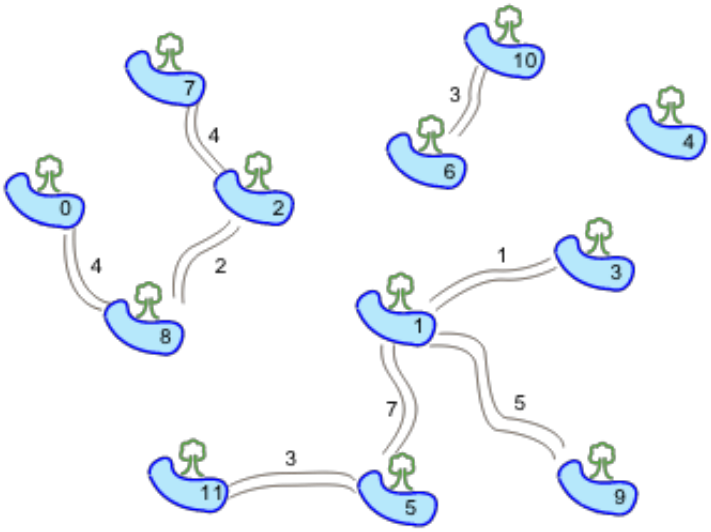
Ova priča smeštena je nekada davno, kad je svet tek nastajao, a IOI nije postojao ni u snovima.

Zmija živi u zemlji u kojoj ima N bazena (rupa sa vodom) označenih brojevima $0, \ldots, N-1$. Postoji M dvosmernih *staza* između parova bazena, po kojima se zmija može kretati. Svaka dva bazena spojena su (direktno ili indirektno) najviše jednim putem (nizom staza), mada određeni parovi bazena uopšte ne moraju biti spojeni (odnosno $M \leq N-1$). Zmiji je potreban određeni broj dana da pređe neku stazu, a ovaj broj može biti različit za svaku stazu.

Zmijin prijatelj, Kengur, želi da napravi $N-M-1$ novih staza tako da se Zmija može kretati između bilo koja dva bazena. Kengur može da napravi stazu između bilo koja dva bazena, a Zmiji će trebati L dana da pređe svaku novu stazu.

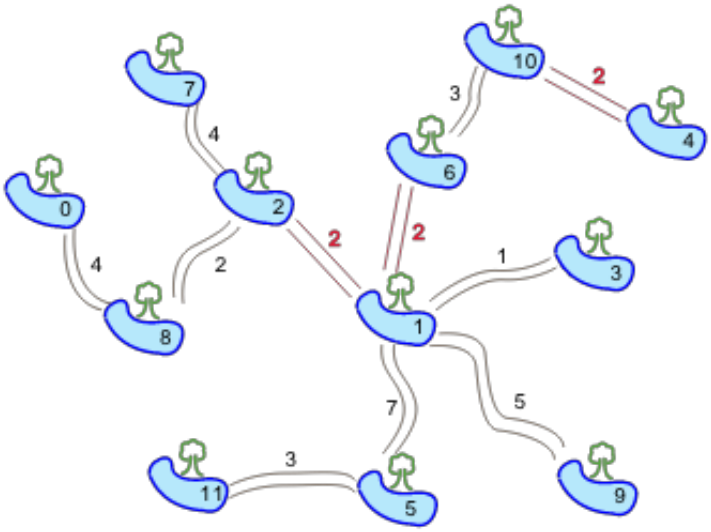
Pored toga, Kengur želi da skrati Zmijina putovanja koliko god je to moguće. Kengur pravi nove staze tako da je najduže vreme putovanja između bilo koja dva bazena što je moguće kraće. Pomozite Kenguru i Zmiji da odrede koliko je ovo najduže vreme putovanja nakon što Kengur napravi nove staze na opisani način.

Primeri



Na slici iznad ima 12 (`N = 12`) bazena i 8 (`M = 8`) staza. Neka je `L = 2` , odnosno Zmiji su potrebna dva dana da pređe svaku novu stazu. Tada bi Kengur mogao odlučiti da napravi tri nove staze:

- između bazena 1 i 2;
- između bazena 1 i 6;
- između bazena 4 i 10.



Na slici iznad prikazan je konačni skup staza. Najduže vreme putovanja je 18 dana (između bazena 0 i 11). Ovo je najbolji mogući rezultat—bez obzira na koji način Kengur izgradi nove staze, postojaće neka dva bazena između kojih će Zmija putovati barem 18 dana.

Implementacija

Vi treba da priložite datoteku koja sadeži implementaciju za funkciju `travelTime()` koja izvršava sledeću radnju:

Vaša funkcija: `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;
```

Opis

Ova funkcija treba da izračuna najduže vreme putovanja (izraženo u danima) između bilo koja dva bazena, uz pretpostavku da je Kengur izgradio `N - M - 1` novih staza tako da su svi bazeni povezani i da najduže vreme putovanja što je moguće manje.

Argumenti (parametri)

- `N`: Broj bazena.
- `M`: Broj staza koje već postoje.
- `L`: Vreme u danima potrebno Zmiji da pređe novu stazu.
- `A`, `B` i `T`: Nizovi dužine `M` koji određuju krajeve već postojećih staza i vreme putovanja po svakoj od njih, tako da `i`-ta staza spaja bazene `A[i-1]` i `B[i-1]`, i da je Zmiji potrebno `T[i-1]` dana da pređe stazu u bilo kom smeru.
- *Vraća*: Najduže vreme putovanja između bilo koja dva bazena, nakon dodavanja staza, kao što je opisano pre.

Primer

Sedeći primer izvršavanja odnosi se na prethodno navedeni primer:

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`

Ograničenja

- Vremensko ograničenje: 1 sekunda
- Memorijsko ograničenje: 64 MiB
- $1 \le N \le 100,000$
- $0 \le M \le N - 1$
- $0 \le A[i], B[i] \le N - 1$
- $1 \le T[i] \le 10,000$
- $1 \le L \le 10,000$

Podzadaci

Podzadatak	Poeni	Dodatna ograničenja na ulazne podatke
1	14	`M = N - 2`, i postoji tačno jedna ili tačno dve postojeće staze koje vode iz svakog od bazena. Drugim rečima, postoje dva skupa povezanih bazena, i u svakom od njih staze formiraju put koji se nikada ne grana.
2	10	`M = N - 2` i `N ≤ 100`
3	23	`M = N - 2`
4	18	Od svakog bazena vodi najviše jedna postojeća staza.
5	12	`N ≤ 3,000`
6	23	*(Bez dodatnih ograničenja)*

Lokalno testiranje

Grejder koji je dostupan na Vašem računaru čita ulazne podatke iz datoteke `dreaming.in`, koja ima sledeći formatu:

- red 1: `N M L`
- redovi 2, …, `M + 1`: `A[i] B[i] T[i]`

Ilustracije radi, ranije opisani primer je zapisan u sledećem formatu:

```
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
```

Napomene za programske jezike

C/C++	Morate dodati `#include "dreaming.h"`.
Pascal	Morate definisati `unit Dreaming`. Svi nizovi su indeksirani od `0` (a ne `1`).

Pogledajte template rešenja na vašem računaru.