



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

dreaming

Lithuanian – 1.0

Ši istorija nutiko labai seniai, kai pasaulis buvo tik susikūręs ir apie IOI net nebuvo sapnuojama.

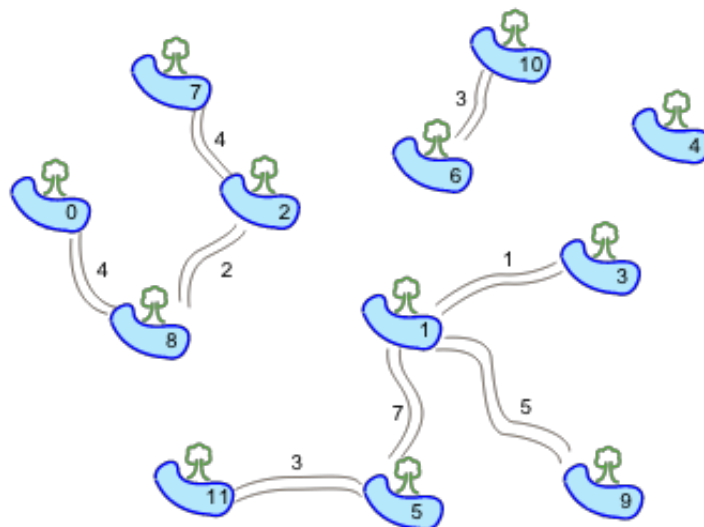
Viename krašte buvo N senvagių (kilpos formos ežerų), sunumeruotų nuo 0 iki $N - 1$. Senvagių poras jungia M dvikrypčių *takų*, kuriais gali šliaužti gyvatė. Bet kuri senvagių pora tiesiogiai ar netiesiogiai yra sujungta ne daugiau kaip viena takų seka. Kai kurios senvagių poros gali būti nesujungtos, taigi $M \leq N - 1$. Kiekvieną taką gyvatė įveikia (peršliaužia) per tam tikrą skaičių dienų. Skirtingiems takams dienų skaičius gali būti skirtingas.

Gyvatės draugė kengūra nori išminti $N - M - 1$ naujų takų taip, kad gyvatė galėtų iš bet kurios senvagės nušliaužti į bet kurią kitą senvagę. Kengūra gali išminti taką tarp bet kurių dviejų senvagių. Kiekvieną naujai kengūros išmintą taką gyvatė peršliauš per L dienų.

Kengūra nori, kad gyvatė galėtų kuo greičiau keliauti. Naujus takus kengūra išmins taip, kad ilgiausias šliaužimo tarp bet kurių dviejų senvagių laikas būtų kuo trumpesnis.

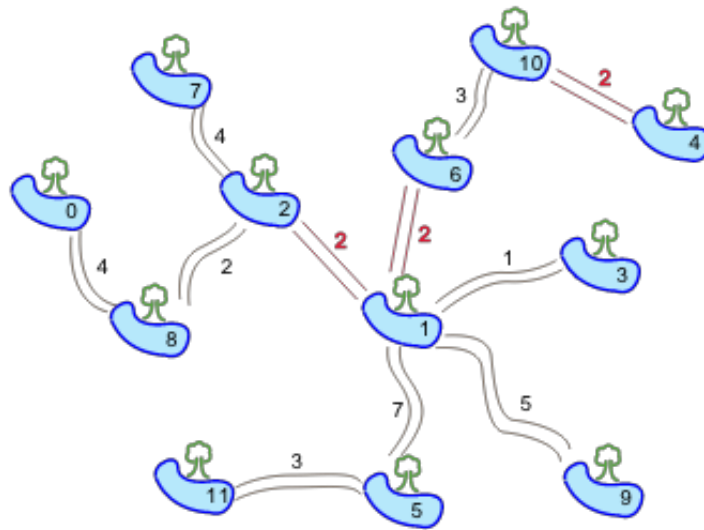
Padėkite kengūrai ir gyvatei suskaičiuoti, koks bus ilgiausias šliaužimo laikas tarp dviejų senvagių.

Pavyzdžiai



Paveiksle yra $N = 12$ senvagių ir $M = 8$ takai. Tarkime, kad $L = 2$, t.y. bet kurį naujai išmintą taką gyvatė nušliauš per 2 dienas. Tokioje situacijoje kengūra gali išminti tris takus:

- tarp 1-os ir 2-os senvagių;
- tarp 1-os ir 6-os senvagių;
- tarp 4-os ir 10-os senvagių.



Paveiksle pateiktas galutinis takų rinkinys. Ilgiausias šliaužimo laikas yra 18 dienų, tarp 0-ės ir 11-os senvagių. Tai yra mažiausias galimas rezultatas — nesvarbu kokius takus kengūra nuspręstų išminti, visuomet egzistuos tokia senvagių pora, tarp kurių gyvatė šliauš 18 ar daugiau dienų.

Realizavimas

Pateikite failą, realizuojantį žemiau apibūdintą funkciją `travelTime()`.

Jūsų 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;
```

Veikimas

Ši funkcija turi apskaičiuoti, kiek ilgiausiai gyvatė šliaužtų tarp dviejų senvagių, jeigu žinoma, kad kengūra išmynė $N - M - 1$ naujų takų taip kaip aprašyta aukščiau.

Parametrai

- `N` : senvagių skaičius.

- M : pradinis takų skaičius.
- L : dienų skaičius, kurio gyvatei prireiks peršliaužti bet kurį naujai išmintą taką.
- A ir B : M dydžio masyvai, nurodantys, kurias senvages jungia takai pradiniu momentu; i -asis takas jungia senvages $A[i-1]$ ir $B[i-1]$.
- T : M dydžio masyvas, nurodantis keliavimo takais trukmę; i -uoju taku gyvatė šliaužtų $T[i-1]$ dienų.
- *return*: ilgiausias keliavimo laikas tarp dviejų senvagių.

Pavyzdžiai

Žemiau pateikiama sesija kurioje naudojamas aukščiau pateiktas pavyzdys:

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

Ribojimai

- Laiko ribojimas: 1 sekundė
- Atminties ribojimas: 64 MiB (MiB ir MB reiškia tą patį)
- $1 \leq N \leq 100,000$
- $0 \leq M \leq N - 1$
- $0 \leq A[i], B[i] \leq N - 1$
- $1 \leq T[i] \leq 10,000$
- $1 \leq L \leq 10,000$

Dalinės užduotys

Dalinė užduotis	Taškai	Papildomi duomenų ribojimai
1	14	$M = N - 2$; pradiniu momentu veda lygiai vienas arba du takai iš kiekvienos senvagės; kitaip sakant, visi takai sudaro du atskirus kelius be išsišakojimų (ir nėra atskirtų senvagių).
2	10	$M = N - 2$ and $N \leq 100$
3	23	$M = N - 2$
4	18	Iš kiekvienos senvagės veda ne daugiau kaip vienas takas.
5	12	$N \leq 3,000$
6	23	<i>Nėra</i>

Eksperimentavimas

Jūsų kompiuteryje įrašytas pavyzdinis vertintojas duomenis skaitys iš failo `dreaming.in`, kurio fomatas turi būti toks:

- 1-a eilutė: `N M L`
- 2-a, ..., $(M + 1)$ -a eilutės: `A[i] B[i] T[i]`

Pavyzdžiui, aukščiau pateiktas pavyzdys turi būti užrašytas tokiu 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
```

Pastabos apie programavimo kalbas

C/C++ Įtraukite `#include "dreaming.h"`.

Pascal Apibrėžkite `unit Dreaming`. Visi masyvai numeruojami nuo `0` (ne nuo `1`).

Naudokite pateiktą sprendimo šabloną kaip pavyzdį.