



International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

dreaming

magyar — 1.0

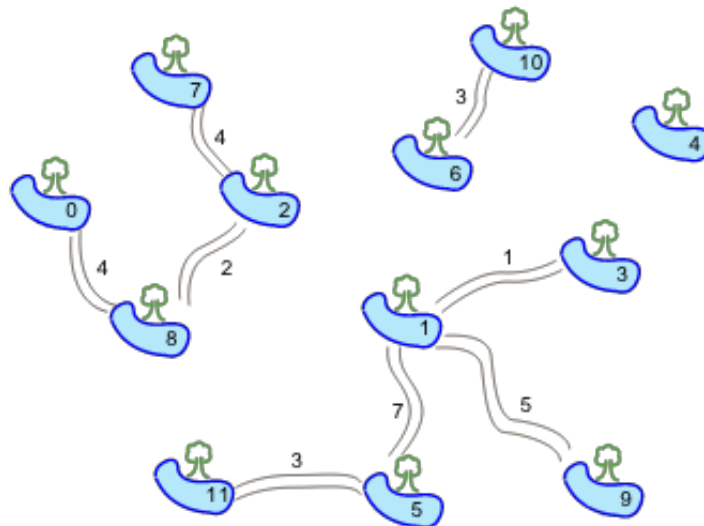
Ez egy régi történet.

A kígyó földjén N holtág (vízzel teli gödör), van, $0, \dots, N-1$ közötti sorszámokkal azonosítva. M kétirányú ösvény, köt össze gödörpárokat, amelyeken a kígyó közlekedhet. Bármely két gödör között legfeljebb 1 útvonal lehet. Lehetnek olyan gödörpárok, amelyek között nincs útvonal (azaz, $M \leq N-1$). Az ösvények hossza azon napok száma, ami alatt a kígyó átjuthat az egyik gödörből a másikba. Ez ösvényenként különböző lehet

A kenguru $N - M - 1$ új ösvényt készít, hogy a kígyónak bármely két gödör között legyen útvonala. A kenguru bármely két gödör között készíthet ösvényt, s mindegyik új ösvény hossza L nap.

A kenguru azt szeretné, hogy a kígyó a lehető leggyorsabban utazzon. Azaz úgy készíti új ösvényeket, hogy a leghosszabb útvonal hossza bármely két gödör között a lehető legkisebb legyen. Meg kell adni a leghosszabb utazási időt az új ösvények elkészítése után!

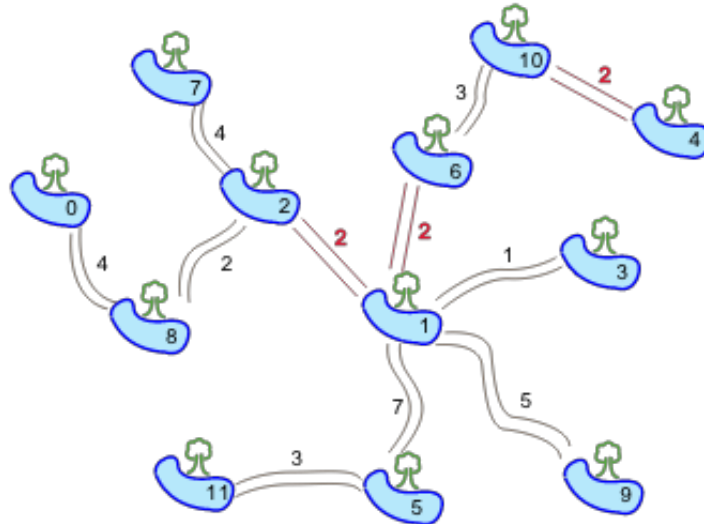
Példák



A fenti képen $N = 12$ gödör és $M = 8$ ösvény van. Az $L = 2$, azaz az új utak hossza 2 nap. A kenguru 3 új ösvényt készít:

- az 1. és a 2. gödör között;
- az 1. és a 6. gödör között;

- a 4. és a 10. gödör között.



A fenti kép a végső állapotot mutatja. A leghosszabb út 18 nap, amely a 0. és a 11. gödör között van. Ennél kevesebbet semmi más módon nem lehet elérni.

Megvalósítás

A `travelTime()` függvényt kell beküldened, a következő formában:

A függvényed: `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;
```

Leírás

A függvény számítsa ki a leghosszabb útvonal hosszát tetszőleges két gödör között, ha a kenguru $N - M - 1$ új ösvényt készített úgy, hogy ez az érték a lehető legkisebb legyen!

Paraméterek

- N : a gödrök száma.
- M : a létező ösvények száma.
- L : az új ösvények hossza napokban.
- A , B és T : M elemű tömbök, az ösvények végpontjai és hossza. Az i . ösvény az $A[i-1]$ és a $B[i-1]$ gödröt köti össze, hossza $T[i-1]$ nap mindkét irányban.

- *Visszatérési érték:* A fentieknek megfelelő leghosszabb útvonal hossza.

Mintapélda

Az ábrának megfelelő bemenet:

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

Korlátok

- Időlimit: 1 másodperc
- Memória limit: 64 MiB
- $1 \leq N \leq 100,000$
- $0 \leq M \leq N - 1$
- $0 \leq A[i], B[i] \leq N - 1$
- $1 \leq T[i] \leq 10,000$
- $1 \leq L \leq 10,000$

Részfeladatok

Részfeladat	Pontszám	További feltételek
1	14	$M = N - 2$, és bármely gödörtől egy vagy kettő ösvény indul. Más szóval az ösvények két láncot alkotnak.
2	10	$M = N - 2$ és $N \leq 100$
3	23	$M = N - 2$
4	18	Minden gödörtől legfeljebb 1 ösvény indul.
5	12	$N \leq 3,000$
6	23	(None)

Gyakorlás

A mintaértékelő a `dreaming.in` fájlt olvassa, ami a következő formátumú:

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

A példa bemenet a fájlban a következő formátumú

```
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
```

Nyelvi előírások

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.