



International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

dreaming

Macedonian — 1.0

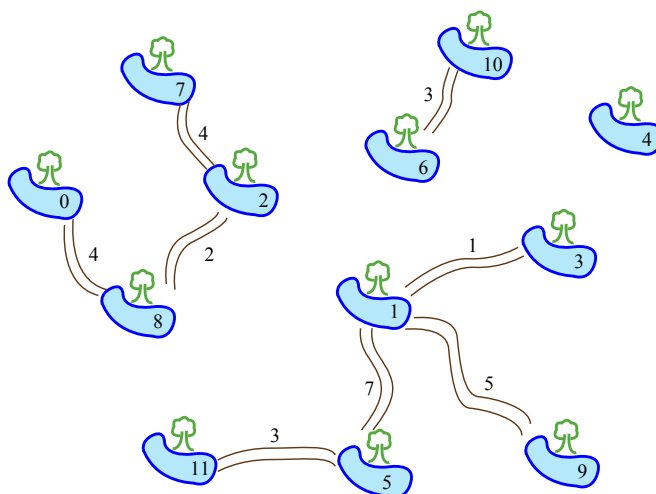
Ова приказна е за бог знае кога одамна.

Ристо живее во земја со N вирчиња (дупки со вода), нумерирани со $0, \dots, N - 1$. постојат M двонасочни *патеки*, кои поврзуваат по две вирчиња (пар од вирчиња) по кои Ристо може да патува. Секој пар од вирчиња е поврзан (директно или индиректно) од најмногу една низа од патеки, иако некои вирчиња може воопшто да не се поврзани (така, $M \leq N - 1$). За да ја помине секоја патека Ристо троши по одреден број на денови: овој број може да е различен за секоја патека.

Блажо, другарот на Ристо, сака да направи $N - M - 1$ нови патеки, така што ќе биде можно Ристо да патува помеѓу кој било пар од вирчиња. Блажо може да направи патека меѓу кои било вирчиња и за секоја така направена патека на Ристо ќе му бидат потребни L денови за да ја помине.

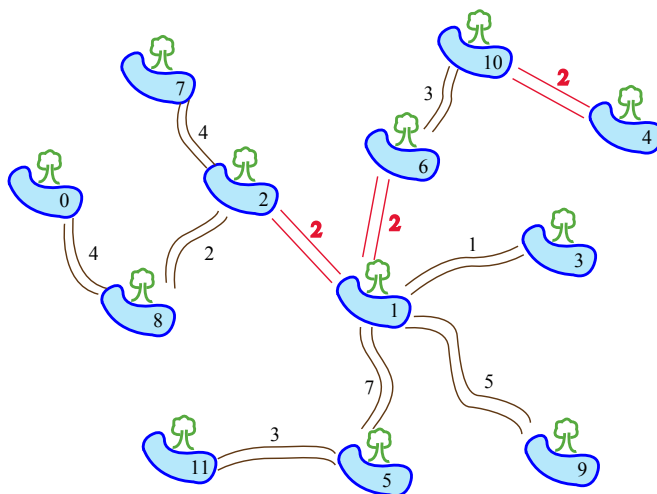
Уште повеќе, Блажо сака Ристо да патува што побрзо (да не ослабне случајно). Блажо ќе изгради нови патеки така што најдолгото време за патување помеѓу две вирчиња е најкратко можно. Помогни им на Блажо и Ристо, да го одредат најдолгото потребно време за патување помеѓу две вирчиња, откако Блажо ги направил новите патеки, разбираш.

Примери



На погорната слика има $N = 12$ вирчиња и $M = 8$ патеки. Претпостава дека $L = 2$, така што за поминување на новоизградените патеки ќе бидат потребни 2 дена. Потоа Блажо може да изгради 3 нови патеки:

- помеѓу вирчињата 1 и 2;
- помеѓу вирчињата 1 и 6;
- помеѓу вирчињата 4 и 10.



Погорната слика го прикажува конечното множество на патеки. Најдолгото време за патување е 18 дена, помеѓу вирчињата 0 и 11. Ова е најмалиот можен резултат - без разлика како Блажо ќе ги изгради патеките, ќе постои пар вирчиња кој побарува Ристо да патува 18 или повеќе денови.

Имплементација

Треба да пратите фајл кој ја имплементира функцијата `travelTime()`, вака:

Вашата функција: `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;
```

Опис

Оваа функција треба да го пресмета најдолгото време за патување (измерено во денови) помеѓу кој било пар вирчиња, под претпоставка дека Блажо додал $N - M - 1$ патеки, и дека бил паметен да ги додаде тие кое овозможуваат најдолгото време за патување да е најмало можно.

Параметри

- N : Бројот на вирчиња.
- M : Бројот на постојни патеки.
- L : Време во денови, кои ќе му се потребни на Ристо да ги помине новододадените патеки.
- A , B и T : Низи со должина M кои ги специфицираат крајните точни и должината на патување за секоја од постојните патеки, така што i -тата патека ги спојува вирчињата $A[i-1]$ и $B[i-1]$, и побарува $T[i-1]$ денови за да се измине (во секоја од двете насоки).
- *Returns*: Најдолгото време за патување помеѓу пар вирчиња, според описот погоре.

Пробна сесија

Следната сесија го објаснува погорниот пример:

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

Ограничувања

- Временско ограничување: 1 секунда
- Мемориско ограничување: 64 MiB
- $1 \leq N \leq 100,000$
- $0 \leq M \leq N - 1$
- $0 \leq A[i], B[i] \leq N - 1$
- $1 \leq T[i] \leq 10,000$
- $1 \leq L \leq 10,000$

Подзадачи

Подзадача	Поени	Дополнителни ограничувања на влезот
1	14	$M = N - 2$, и постојат точно 1 или 2 постојни патеки кои излегуваат од секое вирче. Со други зборови, има две множества од поврзани вирчиња, и во секое множество патеките формираат рамен (неразгранет) пат.
2	10	$M = N - 2$ и $N \leq 100$
3	23	$M = N - 2$
4	18	Постои најмногу еден постоен пат кој излегува од секое вирче.
5	12	$N \leq 3,000$
6	23	(None)

Експериментирање

Пробниот оценувач на твојот компјутер ќе чита влез од фајлот `dreaming.in`, кој мора да е во следниот формат:

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

На пример, погорниот пример треба да е даден во следниот формат:

```
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
```

Забелешки за јазикот

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.