



International Olympiad in Informatics

2013

6-13 July 2013

Brisbane, Australia

dreaming

Azeri — 1.0

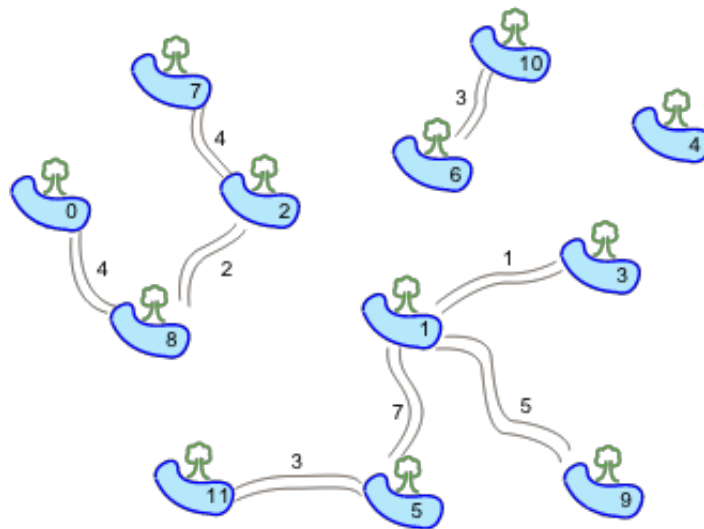
Bu əhvalat çox-çox qədimlərdə – dünya hələ təzə yarananda və IOI hələ xəyallarda belə olmayanda baş verib.

Serpent'in yaşadığı yerdə $0, \dots, N-1$ kimi nömrələnmiş N billabonq (gölməçə) var idi. İki gölməçəni birləşdirən M sayda ikiistiqaətli cığır var idi ki, Serpent həmin cığırlar boyu gəzə bilərdi. Hər bir gölməçə cütü birdən çox cığırlar ardıcılığı ilə birləşib (birbaşa, yaxud dolay), ancaq bəzi gölməçə cütü qalanlarla birləşməyə də bilər (belə ki, $M \leq N-1$). Serpent hər bir cığır müəyyən gün ərzində keçə bilər: bu ədədlər hər bir cığır üçün fərqli ola bilər.

Serpent'in dostu Kenquru $N - M - 1$ sayda yeni cığır açmaq istəyir ki, Serpent istənilən iki gölməçə arasında gəzişə bilsin. Kenquru istənilən iki gölməçə arasında cığır açmağa və Kenqurunun düzəltdiyi hər bir cığırın getməsi üçün Serpentə L gün lazımdır.

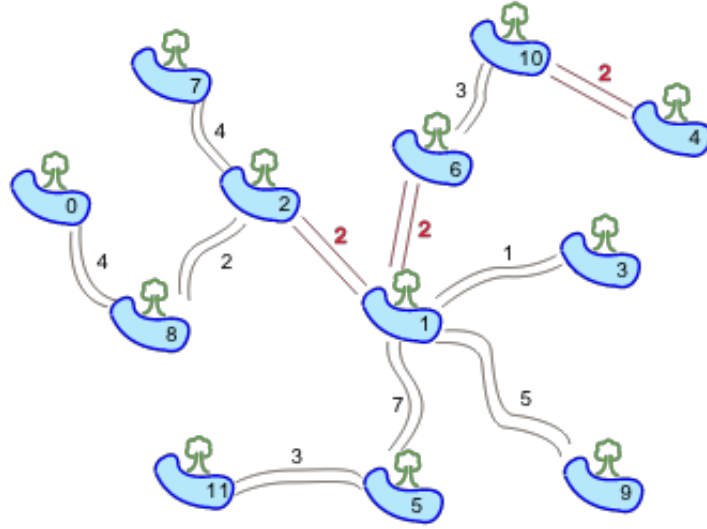
Bundan başqa, Kenquru istəyir ki, Serpent'in hər gəzintisi mümkün qədər az vaxta başa gəlsin. Kenquru o vaxtadək yeni cığırlar açacaq ki, istənilən iki gölməçə arasındakı ən uzun gediş zamanı mümkün qədər kiçik olacaq. İstənilən iki gölməçə arasındakı ən uzun gediş zamanını müəyyən etməkdə Kenquru və Serpentə yardım edin, bundan sonra Kenquru bu yol boyunca yeni cığırlar açacaq.

Örnəklər



Yuxarıdakı şəkildə $N = 12$ gölməçə və $M = 8$ cığır var. Tutaq ki, $L = 2$, yəni istənilən cığırın Serpent 2 günə gedir. Bundan sonra Kenquru üç yeni cığır açmağa bilər:

- 1 və 2 gölməçələri arasında;
- 1 və 6 gölməçələri arasında;
- 4 və 10 gölməçələri arasında.



Yuxarıdakı şəkildə sonda alınan cığırılar çoxluğu göstərilib. Ən uzun gediş zamanı 0 və 11 gölməçələrinin arasında 18 günə başa gəlir. Bu, mümkün olan ən kiçik nəticədir; Kenqurunun cığıruları necə çəkməsinin fərqi yoxdur, elə iki gölməçə olacaq ki, Serpend onlar arındakı cığıruları 18 və ya daha artıq günə gedə biləcək.

Reallalaşdırma

Siz reallaşdırma faylına aşağıdakı şəkildə olan `travelTime()` funksiyasını verməlisiniz:

Sizin funksiya: `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;
```

Təsviri

Bu funksiya istənilən iki gölməçə arasındakı ən uzun gediş zamanını (günlə ölçülən) hesablamalıdır; fərz edilir ki, Kenquru $N - M - 1$ sayda cığır açıb ki, onlar vasitəsilə bütün gölməçələr birləşib və bu ən uzun gediş zamanı mümkün olan ən kiçik zamandır.

Parametrlər

- `N`: Gölməçələrin sayı.

- M : Artıq mövcud çıxırların sayı.
- L : Serpentin yeni çıxırını getməsi üçün tələb olunan günlərin sayı.
- A , B və T : Uzunluqları M olan bu massivlər qabaqcadan mövcud olan çıxırların uc nöqtələrini və gediş zamanını göstərir. Belə ki, i -ci çıxır $A[i-1]$ və $B[i-1]$ gölməçələrini birləşdirir və onların birindən obirinə gediş zamanı $T[i-1]$ gün çəkir.
- *Qaytarılan qiymətlər*: İstənilən iki gölməçə arasında ən böyük gediş zamanı.

Sample Session

Aşağıdakı sessiya yuxarıda verilmiş örnəyi təsvir edir:

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

Məhdudiyyətlər

- Zaman limiti: 1 saniyə
- Yaddaş limiti: 64 MiB
- $1 \leq N \leq 100,000$
- $0 \leq M \leq N - 1$
- $0 \leq A[i], B[i] \leq N - 1$
- $1 \leq T[i] \leq 10,000$
- $1 \leq L \leq 10,000$

Altməsələlər

Altməsələ	Ballar	Giriş verilənlərinə əlavə məhdudiyyətlər
1	14	$M = N - 2$ və hər bir gölməçədən başlayan əvvəldən mövcud olan düz bir, yaxud iki cığır var. Başqa sözlə, birləşmiş gölməçələrin iki çoxluğu var və hər bir çoxluqda cığırlar budaqlanmayan yol əmələ gətirir.
2	10	$M = N - 2$ və $N \leq 100$
3	23	$M = N - 2$
4	18	Hər bir gölməçədən başlayan qabaqcadan mövcud olan ən çoxu bir cığır var.
5	12	$N \leq 3,000$
6	23	(None)

Sınaqdan keçirmə

Kompüterinizdə olan qiymətləndirmə sistemi nümunəsi aşağıdakı formatda olacaq giriş verilənlərini `dreaming.in` faylından oxuyacaq:

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

Məsələn, yuxarıdakı nümunə aşağıdakı formatda verilməlidir:

```
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
```

Dil qeydləri

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.