



International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

dreaming

Polish — 1.0

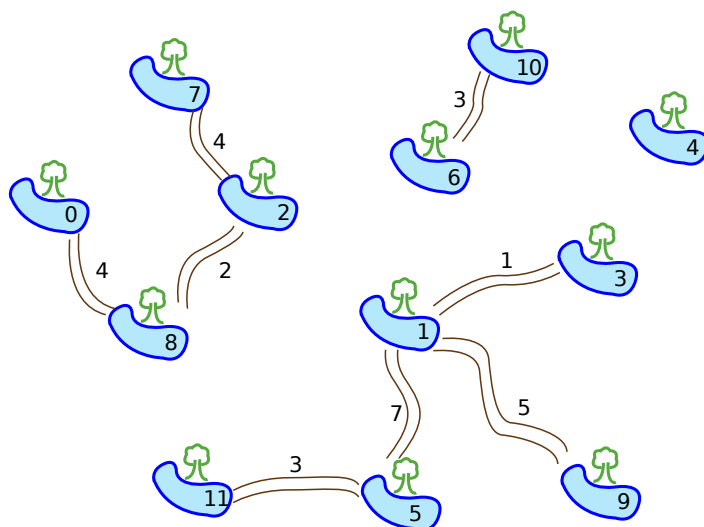
Rzecz dzieje się dawno temu, niedługo po powstaniu świata, w czasach, gdy o IOI jeszcze nikomu się nie śniło.

Pewien Wąż żyje w kraju, w którym znajduje się N kałuż ponumerowanych liczbami $0, \dots, N-1$. Kałuże połączone są za pomocą M dwukierunkowych ścieżek, po których może poruszać się Wąż. Każda para kałuż jest połączona (pośrednio lub bezpośrednio) co najwyżej jednym ciągiem ścieżek. Pewne pary kałuż mogą być ze sobą w ogóle nie połączone (zatem $M \leq N-1$). Przebycie każdej ścieżki zajmuje Wężowi pewną liczbę dni; liczba ta może być różna dla różnych ścieżek.

Kangur, przyjaciel Węża, chciałby wytyczyć (wydeptać) $N-M-1$ nowych ścieżek tak, by Wąż mógł przemieszczać się między dowolnymi dwiema kałużami. Kangur może połączyć dowolne dwie kałuże ścieżką. Przebycie każdej z tych nowych ścieżek będzie zajmowało Wężowi dokładnie L dni.

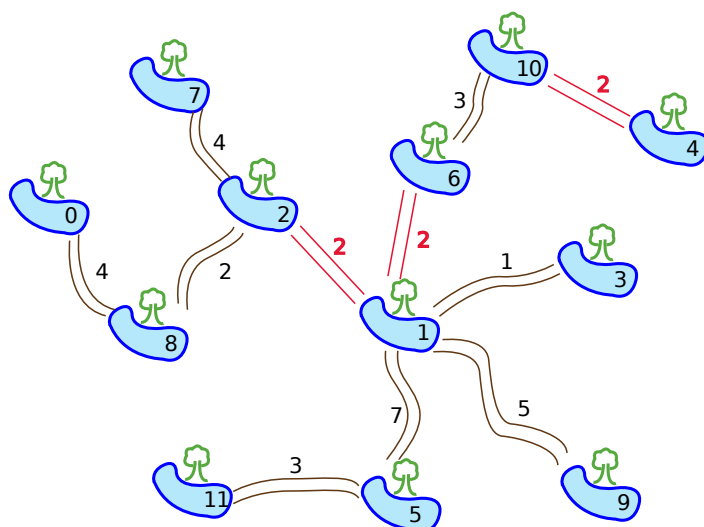
Kangur chciałby, by Wąż przemieszczał się możliwie efektywnie, dlatego wytyczy nowe ścieżki tak, by najdłuższy czas podróży między dowolną parą kałuż był jak najkrótszy. Pomóż zwierzętom wyznaczyć maksymalny czas podróży pomiędzy dowolnymi dwiema kałużami, po tym jak Kangur wydepcze nowe ścieżki.

Przykłady



Powyższy rysunek przedstawia $N = 12$ kałuż i $M = 8$ ścieżek. Przyjmijmy, że $L = 2$, tj. Wąż potrzebuje 2 dni na przebycie każdej nowej ścieżki. W tej sytuacji Kangur może wydeptać np. następujące trzy nowe ścieżki:

- pomiędzy kałużami 1 i 2;
- pomiędzy kałużami 1 i 6;
- pomiędzy kałużami 4 i 10.



Powyższy rysunek przedstawia końcowy układ ścieżek. Najdłuższy możliwy czas podróży między dwiema kałużami to 18 dni – tyle trwa podróż między kałużami 0 i 11. Jest to optymalny wynik, gdyż niezależnie od tego, jak Kangur wydepcze ścieżki, będzie istniała taka para kałuż, że najkrótszy czas podróży Węża pomiędzy tymi kałużami to co najmniej 18 dni.

Implementacja

Powinieneś zgłosić plik z implementacją funkcji `travelTime()`:

Twoja funkcja: `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

Funkcja ta powinna obliczać największy czas podróży (w dniach) pomiędzy dowolną parą kałuż, przy założeniu, że Kangur wydeptał $N-M-1$ ścieżek w taki sposób, że wszystkie kałuże są połączone ścieżkami, a najdłuższy czas podróży pomiędzy dowolną parą kałuż jest jak najmniejszy.

Parametry

- `N` : liczba kałuż
- `M` : liczba istniejących ścieżek
- `L` : liczba dni, których potrzebuje Wąż na przebycie ścieżki wydeptanej przez Kangura.
- `A` , `B` oraz `T` : tablice rozmiaru `M`, które określają końce oraz czasy podróży dla każdej obecnie istniejącej ścieżki; `i`-ta ścieżka łączy kałuże `A[i-1]` oraz `B[i-1]` . Przebycie jej zajmuje `T[i-1]` dni.
- *Wynik funkcji*: Największy czas podróży między dowolną parą kałuż (zgodnie z powyższym opisem).

Przykład

Oto przykładowe parametry funkcji oraz prawidłowy wynik. Sorry że tabelka po angielsku, nie dało się przetłumaczyć:

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

Ograniczenia

- **Maksymalny czas działania: 1 sekunda**
- **Limit pamięci: 64 MiB**
- $1 \leq N \leq 100,000$
- $0 \leq M \leq N - 1$
- $0 \leq A[i], B[i] \leq N - 1$
- $1 \leq T[i] \leq 10,000$
- $1 \leq L \leq 10,000$

Podzadania

Podzadanie	Punkty	Dodatkowe ograniczenia
1	14	$M = N - 2$ i z każdej kałuży wychodzą jedna lub dwie ścieżki. Innymi słowy, kałuże podzielone są na dwa spójne zbiory, a w ramach każdego zbioru są one połączone w graf będący ścieżką prostą.
2	10	$M = N - 2$ i $N \leq 100$
3	23	$M = N - 2$
4	18	Z każdej kałuży wychodzi co najwyżej jedna ścieżka.
5	12	$N \leq 3,000$
6	23	(brak)

Uruchamianie lokalne

Przykładowy moduł oceniający na Twoim komputerze czyta wejście z pliku `dreaming.in` w następującej formie:

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

Dane z powyższego przykładu powinny być więc podane w następującej formie:

```
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
```

Uwagi natury językowej

- | | |
|--------|--|
| C/C++ | Użyj dyrektywy <code>#include "dreaming.h"</code> . |
| Pascal | Musisz zdefiniować <code>unit Dreaming</code> . Wszystkie tablice są indeksowane od <code>0</code> (a nie od <code>1</code>). |

Przykłady znajdziesz w przykładowych rozwiązaniach na Twoim komputerze.