Dreaming

International Olympiad in Informatics 2013, Day 1
6-13 July 2013
Brisbane, Australia

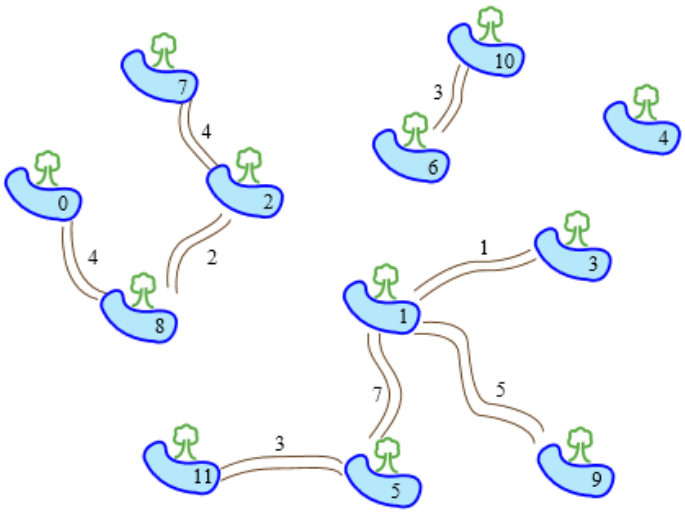
dreaming
Bulgarian — 1.0

Змията живее в страна с N дупки, номерирани от 0 до $N - 1$. Има M двупосочни пътеки, всяка съединяваща по две от дупките. По тези пътеки ще се движи Змията. Всяка двойка дупки е свързана (пряко или непряко) чрез най-много една последователност от пътеки, но може да има двойки дупки, които не са свързани (т.е. $M \leq N - 1$). Времето за преминаване по дадена пътека между две дупки е зададен брой дни и може да е различно при различните двойки дупки, съединени с пътека.

Кенгуруто иска да направи $N - M - 1$ нови пътеки така, че Змията да може да преминава от всяка дупка в всяка друга. Кенгуруто има възможност да направи пътека между всяка двойка дупки. Времето за пътуване на Змията по всяка нова пътека е L дни.

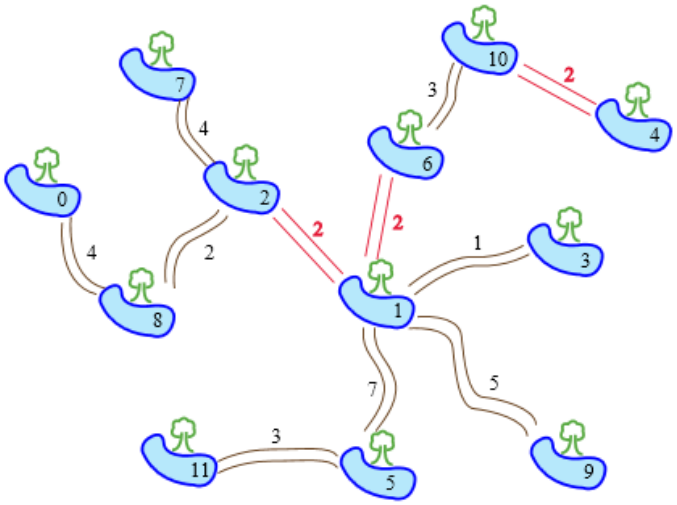
Освен това, Кенгуруто иска Змията да може да се придвижва възможно най-бързо. Кенгуруто ще построи новите пътеки така, че най-дългото време за пътуване между коя да е двойка дупки да е възможно най-малко. Помогнете на Кенгуруто и Змията да намерят най-дългото време за пътуване между двойка дупки, след като Кенгуруто построи новите пътеки.

Пример



Изобразени са N=12 дупки и M=8 пътеки. Нека L=2, т.е. времето за преминаване по всяка от новите пътеки, построени от Кенгуруто е 2 дни. Кенгуруто може да построи следните 3 нови пътеки:

- Между дупка 1 и 2
- Между дупка 1 и 6
- Между дупка 4 и 10



Фигурата показва крайното множество от пътеки. Най-дългото време за пътуване е 18 дни и то е между дупки 0 и 11. Това е най-малкия възможен резултат – по какъвто и начин Кенгуруто да построи новите пътеки, ще има двойка от дупки, разстоянието между които ще бъде преминато от Змията за 18 или повече дни.

Имплементация

Трябва да изпратите файл, съдържащ кода на функция `travelTime()`, която има следното описание:

Вашата функция: `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;
```

Описание:

Функцията трябва да пресметне най-голямото време за пътуване (измерено в дни) между двойка дупки, като е дадено, че Кенгуруто построява $N - M - 1$ пътеки, така че всички дупки се свързват и най-голямото време за пътуване е най-малко.

Параметри:

- `N`: Брой на дупките.
- `M`: Брой на пътеките в началото.
- `L`: Време в дни за преминаване на всяка пътека, която Кенгуруто построява.
- `A`, `B` и `T`: Масиви с дължина `M`, които определят крайните точки и времето за пътуване по съществуващите пътеки, т.е. **i**-тата пътека свързва дупки `A[i-1]` и `B[i-1]` и има време за преминаване `T[i-1]` дни.
- *Returns*: Най-голямото време за пътуване между двойка дупки, както е описано по-горе.

Пример

Параметър	Стойност
`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`

Ограничения:

- Time limit: 1 second
- Memory limit: 64 MiB
- **$1 \le N \le 100{,}000$**
- **$0 \le M \le N - 1$**
- **$0 \le A[i], B[i] \le N - 1$**
- **$1 \le T[i] \le 10{,}000$**
- **$1 \le L \le 10{,}000$**

Подзадачи:

Подзадача	Точки	Допълнителни ограничения за входа
1	14	**M = N – 2** и от всяка дупка излиза точна една или две пътеки, измежду първоначално дадените. С други думи, има две множества от свързани дупки и във всяко от тях пътеките образуват неразклоняващ се път.
2	10	**M = N - 2** and **N ≤ 100**
3	23	**M = N - 2**
4	18	От всяка дупка излиза най-много една пътека от първоначално дадените.
5	12	**N ≤ 3,000**
6	23	*без допълнителни ограничения*

Експериментиране

Опростеният грейдер на вашата машина чете от файла `dreaming.in`, който трябва да е в следния формат:

- ред 1: `N M L`
- редове 2, …, M + 1: `A[i] B[i] T[i]`

Горният пример трябва да се запише по следния начин:

```
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
```

Бележки за езиците за програмиране:

C/C++ Трябва да използвате `#include "dreaming.h"`.

Pascal Дефинирайте `unit Dreaming`. Всички масиви започват от 0.

Вижте темплейта, който се намира на вашата машина.