



## International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

# dreaming

Russian — 1.0

---

Эта история происходила много-много лет назад в те времена, когда о проведении международной олимпиады по информатике никто даже и не мечтал.

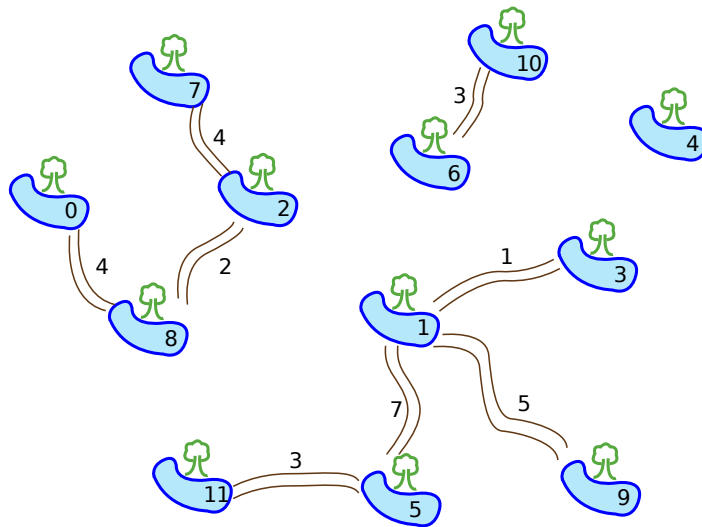
В одной стране живёт Змей. В этой стране есть  $N$  озёр, пронумерованных от  $0$  до  $(N - 1)$ . Кроме того, изначально есть  $M$  двусторонних тропинок, соединяющих пары озёр. По тропинкам может перемещаться Змей. Каждая пара озёр соединена (напрямую или через промежуточные озёра) не более чем одной последовательностью изначально заданных тропинок, хотя какие-то пары озёр могут быть вообще не соединены (таким образом,  $M \leq N - 1$ ). Для каждой изначально заданной тропинки известно количество дней, которое тратит Змей, перемещаясь по этой тропинке. Эти числа могут различаться для разных тропинок.

Друг Змея, Кенгуру, хочет проложить  $(N - M - 1)$  новую тропинку так, чтобы Змей смог путешествовать между любой парой озёр. Кенгуру может создавать новую тропинку между любой парой озёр, и перемещение по каждой новой тропинке, которую создаёт Кенгуру, занимает у Змея  $L$  дней.

Кенгуру хочет сделать путешествия Змея как можно более быстрыми. С этой целью Кенгуру собирается проложить новые тропинки таким образом, чтобы максимальное время путешествия между двумя озёрами было как можно меньше. Помогите Кенгуру и Змею определить максимальное время путешествия между двумя озёрами после того, как Кенгуру проложит новые тропинки указанным способом.

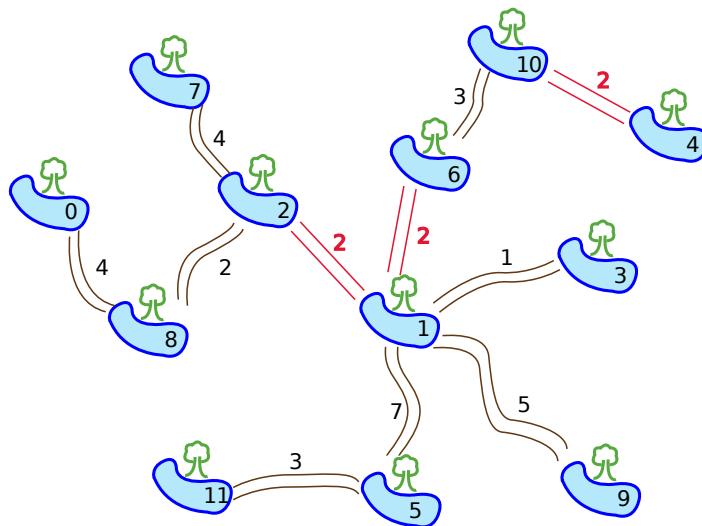
---

## Примеры



На рисунке выше показаны  $N = 12$  озёр и  $M = 8$  изначально заданных тропинок. Предположим, что  $L = 2$ , то есть время перемещения по каждой из новых тропинок занимает 2 дня. Тогда Кенгуру может построить такие три новых тропинки:

- между озёрами 1 и 2;
- между озёрами 1 и 6;
- между озёрами 4 и 10.



На рисунке выше показан окончательный набор тропинок. Максимальное время путешествия — 18 дней, оно достигается при путешествии между озёрами 0 и 11. Это наименьшее возможное максимальное время путешествия: как бы Кенгуру ни строил новые тропинки, всегда будет находиться какая-то пара озёр, для путешествия между которыми потребуется 18 дней или более.

---

## Детали реализации

Ваше решение должно реализовывать нижеописанную функцию `travelTime()` :

**Ваша 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

Эта функция должна вычислять максимальное время (в днях), которое может потребоваться для путешествия от одного озера до другого, в предположении, что Кенгуру добавил  $(N - M - 1)$  новую тропинку таким образом, что все озёра связаны между собой, и это максимальное время является минимально возможным.

### Параметры

- `N` : количество озёр.
- `M` : количество изначально заданных тропинок.
- `L` : время в днях, которое потребуется Змею на перемещение по любой из проложенных новых тропинок.
- `A`, `B` и `T` : массивы длины `M`, которые задают концы и время перемещения по каждой из изначально заданных тропинок, таким образом, что `i`-я изначально заданная тропинка соединяет озёра `A[i-1]` и `B[i-1]`, а `T[i-1]` дней уходит на путешествие по ней в любом направлении.
- *Возвращаемое значение*: максимальное время путешествия между любой парой озёр, как описано выше.

---

## Пример

Следующий набор данных соответствует вышеприведённому примеру:

| Parameter      | Value                     |
|----------------|---------------------------|
| <b>N</b>       | 12                        |
| <b>M</b>       | 8                         |
| <b>L</b>       | 2                         |
| <b>A</b>       | [0, 8, 2, 5, 5, 1, 1, 10] |
| <b>B</b>       | [8, 2, 7, 11, 1, 3, 9, 6] |
| <b>T</b>       | [4, 2, 4, 3, 7, 1, 5, 3]  |
| <b>Returns</b> | 18                        |

## Ограничения

- Ограничение по времени: 1 секунда
- Ограничение по памяти: 64 МиБ
- $1 \leq N \leq 100\,000$
- $0 \leq M \leq N - 1$
- $0 \leq A[i], B[i] \leq N - 1$
- $1 \leq T[i] \leq 10\,000$
- $1 \leq L \leq 10\,000$

## Подзадачи

| Подзадача | Баллы | Дополнительные ограничения на входные данные   |
|-----------|-------|--|
| 1         | 14    | $M = N - 2$ и из каждого озера выходит ровно одна или две изначально заданные тропинки. Другими словами, есть два набора связанных между собой озёр, в каждом из которых изначально заданные тропинки соединены в неветвящийся путь. |
| 2         | 10    | $M = N - 2$ и $N \leq 100$   |
| 3         | 23    | $M = N - 2$  |
| 4         | 18    | Из каждого озера выходит не более одной изначально заданной тропинки.  |
| 5         | 12    | $N \leq 3\,000$  |
| 6         | 23    | (нет)  |

---

## Взаимодействие с проверяющим модулем

Выданный вам проверяющий модуль на вашем компьютере будет читать ввод из файла `dreaming.in`, который должен иметь следующий формат:

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

В частности, вышеописанный пример должен иметь следующий 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
```

---

## Особенности конкретных языков программирования

- |        |  |
|--------|--|
| C/C++  | Вы должны подключить заголовочный файл с помощью <code>#include "dreaming.h"</code> .  |
| Pascal | Вы должны написать модуль с заголовком <code>unit Dreaming</code> . Все массивы нумеруются, начиная с <code>0</code> (а не с <code>1</code> ). |

Для примера посмотрите шаблоны решений на вашем компьютере.