

## Содержание

Задача А. Предок [1 секунда, 256 мегабайт]	2
Задача В. Самое дешёвое ребро [2 секунды, 256 мегабайт]	3
Задача С. LCA - 2 [3 секунды, 256 мегабайт]	4
Задача D. RMQ [3 секунды, 256 мегабайт]	5
Задача Е. Цветные волшебники [2 секунды, 256 мегабайт]	6
Задача F. Опекуны карнотавров [2 секунды, 64 мегабайта]	7
Задача G. Дерево [4 секунды, 256 мегабайт]	8

---

Во всех задачах работает стандартный ввод-вывод.

### Задача А. Предок [1 секунда, 256 мегабайт]

Напишите программу, которая для двух вершин дерева определяет, является ли одна из них предком другой.

#### Формат входных данных

Первая строка входного файла содержит целое число  $n$  ( $1 \leq n \leq 100\,000$ ) — количество вершин в дереве.

Во второй строке находятся  $n$  чисел,  $i$ -е из которых определяет номер непосредственного родителя вершины с номером  $i$ . Если это число равно нулю, то вершина является корнем дерева.

В третьей строке находится число  $m$  ( $1 \leq m \leq 100\,000$ ) — количество запросов.

Каждая из следующих  $m$  строк содержит два различных числа  $a$  и  $b$  ( $1 \leq a, b \leq n$ ).

#### Формат выходных данных

Для каждого из  $m$  запросов выведите на отдельной строке число 1, если вершина  $a$  является одним из предков вершины  $b$ , и 0 в противном случае.

#### Примеры

ancestor.in	ancestor.out
6	0
0 1 1 2 3 3	1
5	1
4 1	0
1 4	0
3 6	
2 6	
6 5	

**Задача В. Самое дешёвое ребро [2 секунды, 256 мегабайт]**

Дано подвешенное дерево с корнем в первой вершине. Все ребра имеют веса (стоимости). Вам нужно ответить на  $M$  запросов вида «найти у двух вершин минимум среди стоимостей ребер пути между ними».

**Формат входных данных**

В первой строке файла записано одно целое число  $n$  ( $2 \leq n \leq 5 \cdot 10^4$ ) — количество вершин.

В  $i$ -й из следующих  $n - 1$  строк записано два целых числа  $x$  и  $y$  ( $x \leq i, |y| \leq 10^6$ ) — предок вершины  $i$  и стоимость ребра.

Далее следуют  $m$  ( $0 \leq m \leq 5 \cdot 10^4$ ) запросов вида  $(x, y)$  — найти минимум на пути из  $x$  в  $y$  ( $x \neq y$ ).

**Формат выходных данных**

Выведите  $m$  ответов на запросы.

**Примеры**

minonpath.in	minonpath.out
5	2
1 2	2
1 3	
2 5	
3 2	
2	
2 3	
4 5	

### Задача С. LCA - 2 [3 секунды, 256 мегабайт]

Задано подвешенное дерево, содержащее  $n$  вершин, пронумерованных от 0 до  $n - 1$ .

Требуется ответить на  $m$  запросов о наименьшем общем предке для пары вершин.

Запросы генерируются следующим образом. Заданы числа  $a_1, a_2$  и числа  $x, y$  и  $z$ .

Числа  $a_3, \dots, a_{2m}$  генерируются следующим образом:  $a_i = (x \cdot a_{i-2} + y \cdot a_{i-1} + z) \bmod n$ .

Первый запрос имеет вид  $\langle a_1, a_2 \rangle$ . Если ответ на  $i - 1$ -й запрос равен  $v$ , то  $i$ -й запрос имеет вид  $\langle (a_{2i-1} + v) \bmod n, a_{2i} \rangle$ .

#### Формат входных данных

Первая строка содержит два числа:  $n$  ( $1 \leq n \leq 100\,000$ ) и  $m$  ( $1 \leq m \leq 10\,000\,000$ ). Корень дерева имеет номер 0. Вторая строка содержит  $n - 1$  целых чисел,  $i$ -е из этих чисел равно номеру родителя вершины  $i$ .

Третья строка содержит целые числа  $a_1$  и  $a_2$  ( $0 \leq a_i \leq n - 1$ ).

Четвёртая строка содержит три целых числа:  $x, y$  и  $z$  ( $0 \leq x, y, z \leq 10^9$ )

#### Формат выходных данных

Выведите в выходной файл сумму номеров вершин — ответов на все запросы.

#### Примеры

lca2.in	lca2.out
3 2 0 1 2 1 1 1 0	2
1 2 0 0 1 1 1	0

### Задача D. RMQ [3 секунды, 256 мегабайт]

Есть массив из  $N$  целых чисел и  $M$  запросов вида: найдите минимум на отрезке с концами  $l_i, r_i$ .

#### Формат входных данных

Входной файл содержит  $T$  наборов тестовых данных. Каждый набор тестовых данных задаётся числами  $N, M, A, B$  ( $1 \leq N \leq 25\,000, 1 \leq A, B \leq 10^9$ ), где  $N$  — размер массива,  $M$  — число запросов.

Массив и запросы нужно получить следующим образом: выпишем последовательность чисел  $C_i = A \cdot i + B \bmod 2^{32}$ .

Элементы последовательности с номерами от 1 до  $N$  — элементы массива. Элементы последовательности с номерами от  $N + 1$  до  $N + 2 \cdot M$  взятые по модулю  $N$  образуют  $M$  пар чисел, которые являются границами отрезков запросов. Ввод заканчивается числами 0 0 0 0. Массив индексируется с нуля.

Сумма  $N$  по всем наборам тестовых данных не превосходит  $10^8$ . Сумма  $M$  по всем наборам тестовых данных не превосходит  $2 \cdot 10^7$ .

#### Формат выходных данных

Для каждого набора тестовых данных выведите сумму по всем запросам.

#### Примеры

стандартный ввод	стандартный вывод	Пояснение
10 10 955379886 619166003 0 0 0 0	7671393960	массив: 1574545889 2529925775 3485305661 145718251 1101098137 2056478023 3011857909 3967237795 627650385 1583030271  запросы: 7 3 3 9 5 1 7 7 3 9 5 5 1 7 3 9 9 5 1 7

### Задача Е. Цветные волшебники [2 секунды, 256 мегабайт]

Сказочная страна представляет собой множество городов, соединенных дорогами с двухсторонним движением. Причем из любого города страны можно добраться в любой другой город либо непосредственно, либо через другие города. Известно, что в сказочной стране не существует дорог, соединяющих город сам с собой и между любыми двумя разными городами, существует не более одной дороги.

В сказочной стране живут желтый и синий волшебники. Желтый волшебник, пройдя по дороге, перекрашивает ее в желтый цвет, синий — в синий. Как известно, при наложении желтой краски на синюю, либо синей краски на желтую, краски смешиваются и превращаются в краску зеленого цвета, который является самым нелюбимым цветом обоих волшебников.

В этом году в столице страны (городе  $f$ ) проводится конференция волшебников. Поэтому желтый и синий волшебники хотят узнать, какое минимальное количество дорог им придется перекрасить в зеленый цвет, чтобы добраться в столицу. Изначально все дороги не покрашены.

Начальное положение желтого и синего волшебников заранее не известно. Поэтому необходимо решить данную задачу для  $k$  возможных случаев их начальных расположений.

#### Формат входных данных

Первая строка входного файла содержит целые числа:  $n$  ( $1 \leq n \leq 100\,000$ ) и  $m$  ( $1 \leq m \leq 500\,000$ ) — количество городов и дорог в волшебной стране соответственно.

Вторая строка содержит одно целое число  $f$  ( $1 \leq f \leq n$ ) — номер города, являющегося столицей сказочной страны. В следующих  $m$  строках, находится описание дорог страны. В этих  $m$  строк записано по два целых числа  $a_i$  и  $b_i$ , означающих, что существует дорога, соединяющая города  $a_i$  и  $b_i$ . Следующая строка содержит целое число  $k$  ( $1 \leq k \leq 100\,000$ ) — количество возможных начальных расположений волшебников. Далее следуют  $k$  строк, каждая из которых содержит два целых числа — номера городов, в которых изначально находится желтый и синий волшебники соответственно.

#### Формат выходных данных

Для каждого из  $k$  случаев, выведите минимальное количество дорог, которое придется покрасить в зеленый цвет волшебникам для того, чтобы добраться в столицу.

#### Примеры

magic.in	magic.out
6 6	1
1	2
1 2	
2 3	
3 4	
4 2	
4 5	
3 6	
2	
5 6	
6 6	

## Задача F. Опекуны карнотавров [2 секунды, 64 мегабайта]

Карнотавры очень внимательно относятся к заботе о своем потомстве. У каждого динозавра обязательно есть старший динозавр, который его опекает. В случае, если опекуна съедают (к сожалению, в юрский период такое не было редкостью), забота о его подопечных ложится на плечи того, кто опекал съеденного динамозавра. Карнотавры — смертоносные хищники, поэтому их обычаи строго запрещают им драться между собой. Если у них возникает какой-то конфликт, то, чтобы решить его, они обращаются к кому-то из старших, которому доверяют, а доверяют они только тем, кто является их опекуном или опекуном их опекуна и так далее (назовем таких динозавров суперопекунами). Поэтому для того, чтобы решить спор двух карнотавров, нужно найти такого динозавра, который является суперопекуном для них обоих. Разумеется, беспокоить старших по пустякам не стоит, поэтому спорщики стараются найти самого младшего из динозавров, который удовлетворяет этому условию. Если у динозавра возник конфликт с его суперопекуном, то этот суперопекун сам решит проблему. Если у динозавра нелады с самим собой, он должен разобраться с этим самостоятельно, не беспокоя старших. Помогите динозаврам разрешить их споры.

### Формат входных данных

В первой строке содержит целое число  $M$  ( $1 \leq M \leq 200\,000$ ) — количество запросов.

Далее следуют  $M$  запросов, описывающие события:

- $+ v$  — родился новый динозавр и опекунство над ним взял динозавр с номером  $v$ . Родившемуся динозавру нужно присвоить наименьший натуральный номер, который до этого еще никогда не встречался.
- $- v$  — динозавра номер  $v$  съели.
- $? u v$  — у динозавров с номерами  $u$  и  $v$  возник конфликт и вам надо найти им третейского судью.

Изначально есть один прадинозавр номер 1. Гарантируется, что он никогда не будет съеден.

### Формат выходных данных

Для каждого запроса типа «?» в выходной файл нужно вывести на отдельной строке одно число — номер самого молодого динозавра, который может выступить в роли третейского судьи.

### Примеры

carno.in	carno.out
11	1
+ 1	1
+ 1	2
+ 2	2
? 2 3	5
? 1 3	
? 2 4	
+ 4	
+ 4	
- 4	
? 5 6	
? 5 5	

### Задача G. Дерево [4 секунды, 256 мегабайт]

Задано подвешенное дерево, содержащее  $n$  ( $1 \leq n \leq 1\,000\,000$ ) вершин. Каждая вершина покрашена в один из  $n$  цветов. Требуется для каждой вершины  $v$  вычислить количество различных цветов, встречающихся в поддереве с корнем  $v$ .

#### Формат входных данных

В первой строке входного файла задано число  $n$ .

Следующие  $n$  строк описывают вершины, по одной в строке. Описание очередной вершины  $i$  имеет вид  $p_i c_i$ , где  $p_i$  — номер родителя вершины  $i$ , а  $c_i$  — цвет вершины  $i$  ( $1 \leq c_i \leq n$ ). Для корня дерева  $p_i = 0$ .

#### Формат выходных данных

Выведите  $n$  чисел, обозначающих количества различных цветов в поддеревьях с корнями в вершинах  $1, \dots, n$ .

#### Примеры

tree.in	tree.out
5	1 2 3 1 1
2 1	
3 2	
0 3	
3 3	
2 1	