

## Содержание

<b>Задачи</b>	<b>2</b>
Задача 1A. Ancestor. Предок [0.1 sec, 256 mb]	2
Задача 1B. Самое дешевое ребро [0.5 sec, 256 mb]	3
Задача 1C. LCA Problem Revisited [5 sec, 256 mb]	4
Задача 1D. Опекуны карнотавров [2 sec, 256 mb]	5
<b>Бонусные задачи</b>	<b>6</b>
Задача 1E. Дерево [2.7 sec, 256 mb]	6

---

В некоторых задачах большой ввод и вывод, в некоторых других — STL, который активно использует динамическую память (set-ы, map-ы). **Смотрите** как пользоваться быстрым вводом-выводом и переопределять стандартный аллокатор.

## Задачи

### Задача 1А. Ancestor. Предок [0.1 sec, 256 mb]

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

#### Формат входных данных

Первая строка входного файла содержит натуральное число  $n$  ( $1 \leq n \leq 100\,000$ ) — количество вершин в дереве. Во второй строке находится  $n$  чисел. При этом  $i$ -ое число второй строки определяет непосредственного родителя вершины с номером  $i$ . Если номер родителя равен 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.

#### Пример

stdin	stdout
6	0
0 1 1 2 3 3	1
5	1
4 1	0
1 4	0
3 6	
2 6	
6 5	

### Задача 1В. Самое дешевое ребро [0.5 sec, 256 mb]

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 числ —  $n$  (количество вершин).

В следующих  $n - 1$  строках записаны два числа —  $x$  и  $y$ . Число  $x$  на строке  $i$  означает, что  $x$  — предок вершины  $i$ ,  $y$  означает стоимость ребра.  $x < i, |y| \leq 10^6$ .

Далее  $m$  запросов вида  $(x, y)$  — найти минимум на пути из  $x$  в  $y$  ( $x \neq y$ ).

Ограничения:  $2 \leq n \leq 5 \cdot 10^4, 0 \leq m \leq 5 \cdot 10^4$ .

#### Формат выходных данных

$m$  ответов на запросы.

#### Пример

stdin	stdout
5	2
1 2	2
1 3	
2 5	
3 2	
2	
2 3	
4 5	

### Задача 1C. LCA Problem Revisited [5 sec, 256 mb]

Задано подвешенное дерево, содержащее  $n$  ( $1 \leq n \leq 100\,000$ ) вершин, пронумерованных от 0 до  $n - 1$ . Требуется ответить на  $m$  ( $1 \leq m \leq 10\,000\,000$ ) 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:  $n$  и  $m$ . 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 два целых числа в диапазоне от 0 до  $n - 1$ :  $a_1$  и  $a_2$ . Четвертая строка содержит три целых числа:  $x, y$  и  $z$ , эти числа неотрицательны и не превосходят  $10^9$ .

#### Формат выходных данных

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

#### Примеры

stdin	stdout
3 2	2
0 1	
2 1	
1 1 0	

### Задача 1D. Опекуны карнотавров [2 сек, 256 mb]

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

Во входном файле записано число  $M$ , обозначающее количество запросов ( $1 \leq M \leq 200\,000$ ). Далее на отдельных строках следуют  $M$  запросов, обозначающих следующие 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

stdin	stdout
11	1
+ 1	1
+ 1	2
+ 2	2
? 2 3	5
? 1 3	
? 2 4	
+ 4	
+ 4	
- 4	
? 5 6	
? 5 5	

## Бонусные задачи

### Задача 1Е. Дерево [2.7 сек, 256 mb]

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  $n$ . Последующие  $n$  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

stdin	stdout
5	1 2 3 1 1
2 1	
3 2	
0 3	
3 3	
2 1	