

у2018-2-3. Запросы на деревьях

Statement  
is not  
available  
on  
English  
language

А. Двоичные подьемы

2 секунды, 256 мегабайт

Задано подвешенное дерево. Найдите для каждой вершины двоичные подьемы: предков, которые находятся от нее на расстоянии  $2^k$  для какого-либо целого  $k$ .

Входные данные

В первой строке входа задано число  $n$  ( $1 \leq n \leq 10^5$ ) — число вершин дерева. Во второй строке заданы  $n$  чисел  $p_i$ . Число  $p_i$  равно номеру вершины, являющейся предком вершины  $i$  (вершины нумеруются с 1) или нулю, если вершина  $i$  — корень дерева.

Выходные данные

Выведите  $n$  строк. В  $i$ -й строке выведите номер вершины  $i$  и далее после двоеточия список требуемых предков, в порядке увеличения расстояния от  $i$ .

входные данные	
8	
5 8 5 0 4 5 4 1	
выходные данные	
1: 5 4	
2: 8 1 4	
3: 5 4	
4:	
5: 4	
6: 5 4	
7: 4	
8: 1 5	

В. LCA

5 seconds, 256 megabytes

You are given a tree rooted for the first vertex. You need to answer  $m$  queries "find the LCA of two vertices". LCA of two vertices  $u$  and  $v$  in a rooted tree is the most distant from root vertex that lies on both paths from  $u$  and from  $v$  to the root.

Input

The first line of input contains one integer  $n$  — the number of vertices in the tree ( $1 \leq n \leq 2 \cdot 10^5$ ).

Each of next  $n - 1$  lines contains one integer  $x$ . Integer  $x$  on line  $i$  means that  $x$  is a parent of vertex  $i$  ( $x < i$ ).

Then you are given an integer  $m$ .

Then you are given  $m$  ( $0 \leq m \leq 5 \cdot 10^5$ ) queries  $(u, v)$  — find the LCA of vertices  $u$  and  $v$  ( $1 \leq u, v \leq n; u \neq v$ ).

Output

For each query, print the LCA of two vertices in a separate line.

input	
5	
1	
1	
2	
3	
2	
2 3	
4 5	

output

1  
1

input

5  
1  
1  
2  
2  
3  
4 5  
4 2  
3 5

output

2  
2  
1

Statement  
is not  
available  
on  
English  
language

С. Самое дешевое ребро

4 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".

Входные данные

В первой строке задано целое число  $n$  — число вершин в дереве ( $1 \leq n \leq 2 \cdot 10^5$ ).

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

Далее заданы  $m$  ( $0 \leq m \leq 5 \cdot 10^5$ ) запросов вида  $(x, y)$  — найти минимум на пути из  $x$  в  $y$  ( $1 \leq x, y \leq n; x \neq y$ ).

Выходные данные

Выведите ответы на запросы.

входные данные	
5	
1 2	
1 3	
2 5	
3 2	
2	
2 3	
4 5	
выходные данные	
2	
2	

входные данные	
5	
1 1	
1 2	
2 3	
3 4	
2	
1 4	
3 2	

выходные данные

1  
1

Statement  
is not  
available  
on  
English  
language

D. Опекуны карнотавров

2 секунды, 512 мегабайт

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.

Входные данные

Во входном файле записано число  $M$ , обозначающее количество запросов ( $1 \leq M \leq 200000$ ). 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.

Выходные данные

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.

входные данные

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

выходные данные

1  
1  
2  
2  
5

Statement  
is not  
available  
on  
English  
language

E. Трамваи

2 секунды, 256 мегабайт

Правительство небольшого города Мухоловска решило улучшить транспортную ситуацию в своем городе. Для этого была построена сеть трамвайных путей, соединяющая  $n$  трамвайных остановок. Для удобства пассажиров между каждой парой остановок можно было проехать на трамвае. С другой стороны, в целях экономии, проехать между двумя остановками можно было единственным образом. Формально говоря, трамвайная сеть представляет собой дерево с  $n$  вершинами. При этом вершины дерева соответствуют остановкам, а ребра — путям.

Изначально по каждому трамвайному пути проходил хотя бы один трамвайный маршрут. Однако со временем некоторые маршруты оказались отменены, а, следовательно, и некоторые трамвайные пути стали невостребованными. Путь считается невостребованным, если ни один трамвайный маршрут по нему не проходит. С целью экономии средств невостребованные трамвайные пути Мухоловска было решено разобрать.

Ваша задача — написать программу для определения числа невостребованных путей.

Входные данные

Первая строка содержит единственное число  $n$  — количество трамвайных остановок города ( $2 \leq n \leq 100\,000$ ). Каждая из следующих  $n - 1$  строк содержит описание одного трамвайного пути (ребра дерева). Описание состоит из двух чисел  $b$  и  $e$  — номеров остановок, соединенных соответствующим путем. Остановки пронумерованы целыми числами от 1 до  $n$ .

В следующей строке содержится число  $m$  — количество трамвайных маршрутов ( $0 \leq m \leq 100\,000$ ). В каждой из следующих  $m$  строк содержится описание трамвайного маршрута. Описание состоит из двух чисел  $x$  и  $y$  — трамвайный маршрут имеет конечные остановки с номерами  $x$  и  $y$  и проходит по кратчайшему пути между ними ( $x \neq y$ ).

Выходные данные

В выходной файл выведите количество невостребованных трамвайных путей Мухоловска.

входные данные

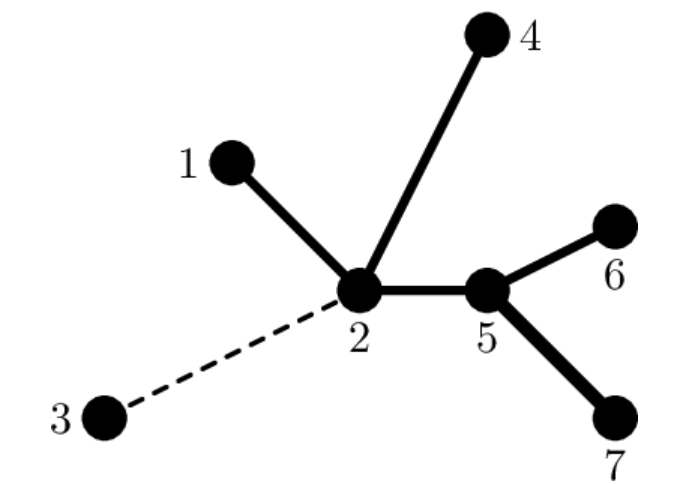
4  
1 2  
1 3  
1 4  
0

выходные данные

3

входные данные
7 1 2 2 3 2 4 5 2 5 6 7 5 3 1 7 2 4 7 6
выходные данные
1

Иллюстрация ко второму примеру.



Пунктирной линией обозначен невостребованный путь.

F. Genealogy

4 seconds, 256 megabytes

During discussions in parliament, lords usually divide into groups with similar views on how to solve the problem. The outcome of the discussion depends on the decision of the most influential group of Lords. That is why the calculation of the influence of the group is the most important task.

Naturally, each lord values the antiquity of a family, therefore the influence of the lord is equal to the antiquity of his family. The antiquity of the lord's family is the number of ancestors of the lord: his father, his grandfather, his great-grandfather, etc. To calculate the influence of a group of lords, you need to calculate the number of lords in the group along with their ancestors. Note that if a lord is the ancestor of two or more lords in a group, then this lord should only be counted once.

You have been given the family tree of lords (surprisingly, all lords are descended from one great-lord) and a list of groups. For each group, find its influence.

Input

First line contains a single integer  $n$  — the number of lords ( $1 \leq n \leq 100\,000$ ). Lords are numbered from 1 to  $n$ . Next line contains  $n$  integers  $p_1, p_2, \dots, p_n$ , where lord number  $p_i$  is a parent of lord number  $i$ . If lord is a founder of a family, then  $p_i$  equals to  $-1$ . It's guaranteed that input forms a tree.

Third line contains a single integer  $g$  — the number of groups ( $1 \leq g \leq 3\,000\,000$ ). Next  $g$  lines contain groups description. Line number  $j$  contains an integer  $k_j$  — the size of  $j$ -th group, followed by  $k_j$  different integers — numbers of lords in  $j$ -th group. It's guaranteed that sum of all  $k_j$  doesn't exceed  $3\,000\,000$ .

Output

Output  $g$  lines. Line number  $j$  should contain the influence of  $j$ -th group. It's guaranteed that size of output doesn't exceed six megabytes.

input
4 -1 1 2 3 4 1 4 2 3 4 3 2 3 4 4 1 2 3 4
output
4 4 4 4

input
5 2 -1 1 2 3 10 3 3 4 1 3 2 4 3 4 1 3 5 4 1 4 2 2 3 3 1 4 3 1 2 3 3 4 5 1 1 3 1 2 4
output
4 4 5 2 3 4 1 5 2 3

G. Path increment

2 seconds, 256 megabytes

You are given a tree. Each vertex has a variable in it. Initially the value of each variable is zero. You need to answer two types of queries: add a number to each vertex on some path, and print the variable.

Input

The first line contains an integer  $n$  — the number of vertices in the tree ( $1 \leq n \leq 3 \cdot 10^5$ ).

Each of the following  $n - 1$  lines contains two integers  $v$  and  $u$  — the edge between  $v$  and  $u$  ( $1 \leq v, u \leq n$ ).

The following line contains an integer  $m$  — the number of queries ( $1 \leq m \leq 5 \cdot 10^5$ ).

The following  $m$  lines describe queries, one query in a line:

- $+ \ v \ u \ d$  — add  $d$  to each variable on a simple path between vertices  $v$  and  $u$  ( $1 \leq v, u \leq n; 1 \leq d \leq 10^9$ );
- $? \ v$  — print the value of variable in vertex  $v$  ( $1 \leq v \leq n$ ).

Output

Print answers to each query.

input
5 1 2 1 3 3 4 3 5 5 + 2 5 1 ? 3 + 1 1 2 ? 1 ? 3

output

1  
3  
1

Statement  
is not  
available  
on  
English  
language

Н. Связность в дереве

2 секунды, 64 мегабайта

Есть граф из  $n$  вершин. Требуется обрабатывать следующие запросы:

- `link U V` — добавить ребро  $UV$ . Гарантируется, что до этого запроса вершины  $U$  и  $V$  были в разных компонентах связности.
- `cut U V` — удалить ребро  $UV$ . Гарантируется, что такое ребро существовало.
- `connected U V` — проверить, правда ли вершины  $U$  и  $V$  лежат в одной компоненте связности.

Входные данные

Первая строка содержит два числа  $n$  ( $2 \leq n \leq 10^5$ ) и  $m$  ( $1 \leq m \leq 10^5$ ) — число вершин и число операций. Следующие  $m$  строк содержат операции.

Выходные данные

Для каждой операции `connected U V` выведите 1, если вершины в одной компоненте или 0 если в разных.

входные данные

5 10  
link 2 5  
link 1 5  
connected 1 2  
cut 2 5  
connected 1 2  
connected 5 1  
link 2 3  
link 2 4  
link 3 5  
connected 1 2

выходные данные

1  
0  
1  
1

Statement  
is not  
available  
on  
English  
language

I. Размер компонент

2 секунды, 64 мегабайта

Есть граф из  $n$  вершин. Требуется обрабатывать следующие запросы:

- `link U V` — добавить ребро  $UV$ . Гарантируется, что до этого запроса вершины  $U$  и  $V$  были в разных компонентах связности.
- `cut U V` — удалить ребро  $UV$ . Гарантируется, что такое ребро существовало.
- `size V` — узнать размер компоненты связности вершины  $V$ .

Входные данные

Первая строка содержит два числа  $n$  ( $2 \leq n \leq 10^5$ ) и  $m$  ( $1 \leq m \leq 10^5$ ) — число вершин и число операций. Следующие  $m$  строк содержат операции.

Выходные данные

Для каждой операции `connected U V` выведите 1, если вершины в одной компоненте или 0 если в разных.

входные данные

5 10  
link 2 5  
link 1 5  
size 1  
cut 2 5  
size 1  
size 2  
link 2 3  
link 2 4  
link 3 5  
size 1

выходные данные

3  
2  
1  
5

J. Decomposition

2 seconds, 256 megabytes

Consider tree  $T$ . Lets call rooted tree  $D(T)$  decomposition tree if it can be built using the following algorithm.

Choose any vertex  $r$  of tree  $T$ . Consider all connected components in tree  $T$  after removing vertex  $r$ :  $S_1, S_2, \dots, S_k$ . Then root of  $D(T)$  is vertex  $r$  and subtrees of vertex  $r$  in  $D(T)$  are  $D(S_1), D(S_2), \dots, D(S_k)$ .

You are given  $T$ . Find its decomposition tree with height at most 20. Height of three is maximum number of vertices in simple path from the root.

Input

First line contains positive integer  $n$  — number of vertices of  $T$  ( $1 \leq n \leq 2 \cdot 10^5$ ).

Next  $n - 1$  lines contain information about edges of the tree. Every edge is pair of integers  $v_i$  and  $u_i$  — ends of this edge ( $1 \leq v_i, u_i \leq n$ ).

Output

Print  $n$  numbers:  $i$ -th number — parent of vertex  $i$  in decomposition tree. If vertex is root of decomposition tree, print 0.

input

3  
1 2  
2 3

output

2 0 2

input

9  
3 2  
4 2  
1 2  
5 1  
1 6  
7 6  
6 8  
8 9

output

0 1 2 2 1 1 6 6 8

Statement  
is not

available  
on  
English  
language

К. Черно-белое дерево

4 секунды, 256 мегабайт

Рассмотрим дерево из  $n$  вершин. Каждая вершина покрашена в черный или белый цвет. Изначально все вершины черные. Требуется отвечать на два типа запросов:

- 1. Поменять цвет вершины.
- 2. Найти сумму расстояний от заданной вершины до всех вершин того же цвета.

Входные данные

Первая строка содержит  $n$  — число вершин дерева ( $1 \leq n \leq 2 \cdot 10^5$ ) и  $m$  — число запросов ( $1 \leq m \leq 10^5$ ).

Следующие  $n - 1$  строк содержат ребра дерева. Каждое ребро описывается парой чисел  $v_i, u_i$  — концы ребра ( $1 \leq v_i, u_i \leq n$ ).

Следующие  $m$  строк содержат запросы, каждый вопрос задается двумя числами: тип запроса (1 или 2) и номер вершины.

Выходные данные

Для каждого запроса второго типа выведите ответ на него.

входные данные
3 3 1 2 2 3 2 1 1 2 2 2
выходные данные
3 0

входные данные
9 5 3 2 4 2 1 2 5 1 1 6 7 6 6 8 8 9 2 1 1 2 2 6 1 5 2 2
выходные данные
14 13 2