

## Лабораторная работа 1-6. Запросы на деревьях

Statement is not available on English language

### А. Двоичные подъемы

ограничение по времени на тест: 2 секунды  
 ограничение по памяти на тест: 256 мегабайт

**ввод: стандартный ввод**

**вывод: стандартный вывод**

Задано подвешенное дерево. Найдите для каждой вершины двоичные подъемы: 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  $i$ .

#### Пример

##### входные данные

Сору

8  
 5 8 5 0 4 5 4 1

##### выходные данные

Сору

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

Statement is not available on English language

### В. LCA

ограничение по времени на тест: 5 секунд  
 ограничение по памяти на тест: 256 мегабайт

**ввод: стандартный ввод**

**вывод: стандартный вывод**

Дано подвешенное дерево с корнем в первой вершине. Вам нужно ответить на  $m$  запросов вида "найти LCA двух вершин". LCA вершин  $u$  и  $v$  в подвешенном дереве — это наиболее удалённая от корня дерева вершина, лежащая на обоих путях от  $u$  и  $v$  до корня.

#### 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 записано одно целое число  $X$ . Число  $X$  на строке  $i$  означает, что  $X$  — предок вершины  $i$  ( $X < i$ ).

Затем дано число  $m$ .

Далее заданы  $m$  ( $0 \leq m \leq 5 \cdot 10^5$ ) запросов вида  $(u, v)$  — найти LCA двух вершин  $u$  и  $v$  ( $1 \leq u, v \leq n$ ;  $u \neq v$ ).

#### Выходные данные

Для каждого запроса выведите LCA двух вершин на отдельной строке.

#### Примеры

##### входные данные

Сору

5  
 1  
 1

2	
3	
2	
2 3	
4 5	
выходные данные	
1	
1	

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

Statement is not available on English language

### С. Самое дешевое ребро

ограничение по времени на тест: 4 секунды

ограничение по памяти на тест: 256 мегабайт

ввод: minonpath.in

вывод: minonpath.out

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.

#### Примеры

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

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

Statement is not available on English language

Statement is not available on English language

## Г. Генеалогия

ограничение по времени на тест: 4 секунды  
ограничение по памяти на тест: 256 мегабайт

**ввод: стандартный ввод**

**вывод: стандартный вывод**

Во время обсуждений в Парламенте лорды, с похожими взглядами на решение проблемы, обычно объединяются в группы. Как правило, результат обсуждения зависит от решения наиболее влиятельной группы лордов. Именно поэтому подсчёт влиятельности группы является наиболее важной задачей.

Естественно, каждый лорд дорожит древностью своего рода, поэтому влиятельность лорда равна древности его рода. Древность рода лорда — количество предков лорда: его отец, его дед, его прадед, и т.д. Чтобы посчитать влиятельность группы лордов, требуется посчитать количество лордов в группе вместе с их предками. Отметим, что если лорд является предком двух или более лордов в группе, то этот лорд должен быть посчитан только один раз.

Вам дано фамильное дерево лордов (удивительно, но все лорды произошли от одного пра-лорда) и список групп. Для каждой группы найдите её влиятельность.

### Входные данные

Первая строка входного файла содержит число  $n$  — количество лордов ( $1 \leq n \leq 100\,000$ ). Лорды нумеруются целыми числами от 1 до  $n$ . Следующая строка содержит  $n$  целых чисел  $p_1, p_2, \dots, p_n$ , где  $p_i$  — отец лорда с номером  $i$ . Если лорд является основателем рода, то  $p_i$  равно -1. Гарантируется, что исходные данные формируют дерево. Третья строка входного файла содержит одно число  $g$  — количество групп ( $1 \leq g \leq 3\,000\,000$ ). Следующие  $g$  строк содержат описания групп.  $j$ -ая строка содержит число  $k_j$  — размер  $j$ -ой группы, после которого следуют  $k_j$  различных чисел — номера лордов, состоящих в  $j$ -ой группе. Гарантируется, что сумма всех  $k_j$  во входном файле не превосходит 3 000 000.

### Выходные данные

В выходной файл выведите  $g$  строк. В  $j$ -ой строке выведите единственное число: влиятельность  $j$ -ой группы. Гарантируется, что размер выходного файла не превосходит шести мегабайт.

### Примеры

#### входные данные

Copy

```
4
-1 1 2 3
4
1 4
2 3 4
3 2 3 4
4 1 2 3 4
```

#### выходные данные

Copy

```
4
4
4
4
```

#### входные данные

Copy

```
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
```

#### выходные данные

Copy

```
4
4
5
```

2  
3  
4  
1  
5  
2  
3

Statement is not available on English language

## Г. Прибавление на пути

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт  
**ввод: стандартный ввод**  
**вывод: стандартный вывод**

Задано дерево. В каждой вершине есть значение, изначально все значения равны нулю. Требуется обработать запрос прибавления на пути и запрос значения в вершине.

### Входные данные

В первой строке задано целое число  $n$  — число вершин в дереве ( $1 \leq n \leq 3 \cdot 10^5$ ).

В следующих  $n - 1$  строках заданы ребра дерева: по два целых числа  $V$  и  $U$  в строке — номера вершин, соединенных ребром ( $1 \leq V, U \leq n$ ).

В следующей строке задано целое число  $m$  — число запросов ( $1 \leq m \leq 5 \cdot 10^5$ ).

Следующие  $m$  строк содержат запросы в одном из двух форматов:

- $+ v \ u \ d$  — прибавить число  $d$  во все значения в вершинах на пути от  $V$  до  $U$  ( $1 \leq V, U \leq n; 1 \leq d \leq 10^9$ );
- $? v$  — вывести значение в вершине  $V$  ( $1 \leq V \leq n$ ).

### Выходные данные

Выведите ответы на все запросы.

### Примеры

#### входные данные

Copy

```
5
1 2
1 3
3 4
3 5
5
+ 2 5 1
? 3
+ 1 1 2
? 1
? 3
```

#### выходные данные

Copy

```
1
3
1
```

Statement is not available on English language

## Н. Связность в дереве

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 64 мегабайта  
**ввод: стандартный ввод**  
**вывод: стандартный вывод**

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

- $\text{link } U \ V$  — добавить ребро  $UV$ . Гарантируется, что до этого запроса вершины  $U$  и  $V$  были в разных компонентах связности.
- $\text{cut } U \ V$  — удалить ребро  $UV$ . Гарантируется, что такое ребро существовало.
- $\text{connected } U \ V$  — 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 V U выведите 1, если вершины в одной компоненте или 0 если в разных.

Пример

входные данные	Copy
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	Copy
1 0 1 1	

Statement is not available on English language

I. Размер компонент

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 64 мегабайта  
ввод: стандартный ввод  
вывод: стандартный вывод

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 V U выведите 1, если вершины в одной компоненте или 0 если в разных.

Пример

входные данные	Copy
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	Copy
3 2 1 5	

J. Decomposition

time limit per test: 2 seconds  
memory limit per test: 256 megabytes  
input: standard input  
output: standard output

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.

Examples

input	Copy
3 1 2 2 3	
output	Copy
2 0 2	

input	Copy
9 3 2 4 2 1 2 5 1 1 6 7 6 6 8 8 9	
output	Copy
0 1 2 2 1 6 6 8	

Statement is not available on English language

К. Черно-белое дерево

ограничение по времени на тест: 4 секунды  
ограничение по памяти на тест: 256 мегабайт  
**ввод: стандартный ввод**  
**вывод: стандартный вывод**

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.

Примеры

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

входные данные	Copy
----------------	------

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

Copy

14  
13  
2