

## Задача А. Декомпозиция

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.6 секунд  
Ограничение по памяти: 256 мегабайт

Рассмотрим дерево  $T$ . Назовем деревом декомпозиции корневое дерево  $D(T)$ . Выберем любую из вершин дерева  $T$ , назовем ее  $r$ . Рассмотрим все компоненты связности дерева  $T$ , после удаления вершины  $r$ :  $S_1, S_2, \dots, S_k$ . Тогда корнем  $D(T)$  будет вершина  $r$ , а детьми  $r$  в  $D(T)$  будут  $D(S_1), D(S_2), \dots, D(S_k)$ .

Вам дано дерево  $T$ . Найдите дерево декомпозиции высоты не более 20. Высота дерева — максимальное число вершин в пути от корня до какой-то вершины.

### Формат входных данных

Первая строка содержит  $n$  ( $1 \leq 2 \cdot 10^5$ ) — количество вершин дерева.

Следующие  $n - 1$  строк содержат пары чисел  $u_i, v_i$  ( $1 \leq u_i, v_i \leq n$ ), описывающие рёбра дерева.

### Формат выходных данных

Выведите  $n$  чисел, где  $i$ -е — родитель вершины  $i$  в дереве декомпозиции. Если вершина — корень, выведите 0.

### Примеры

стандартный ввод	стандартный вывод
3 1 2 2 3	2 0 2
9 3 2 4 2 1 2 5 1 1 6 7 6 6 8 8 9	0 1 2 2 1 1 6 6 8

## Задача В. Найти ближайшую

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Дано дерево из  $n$  вершин, цвет  $i$ -й вершины равен  $a_i$ . Необходимо обработать  $q$  запросов  $(v_i, c_i)$ : найти расстояние от  $v_i$  до ближайшей вершины цвета  $c_i$ . Расстояние между вершинами — минимальное количество рёбер в пути между ними.

### Формат входных данных

Первая строка содержит  $n$  ( $1 \leq n \leq 10^5$ ).

Следующая строка содержит  $n - 1$  число  $p_1, \dots, p_{n-1}$  ( $0 \leq p_i < i$ ).  $p_i$  — отец вершины  $i$ .

Следующая строка содержит числа  $a_1, \dots, a_n$  ( $0 \leq a_i < n$ ).

Следующая строка содержит число  $q$  ( $1 \leq q \leq 10^5$ ).

Следующие  $q$  строк содержат числа  $v_i, c_i$  ( $0 \leq v_i < n, 0 \leq c_i < n$ ).

### Формат выходных данных

Для каждого запроса выведите расстояние до ближайшей вершины требуемого цвета, или  $-1$ , если такой нет.

### Пример

стандартный ввод	стандартный вывод
5	0 1 2 -1 2 1 2 1 1
0 1 1 3	
1 2 3 2 1	
9	
0 1	
0 2	
0 3	
1 0	
2 1	
2 2	
3 3	
3 1	
4 2	

## Задача С. Красим дерево

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.6 секунд  
Ограничение по памяти: 256 мегабайт

Дано взвешенное дерево. Вам необходимо выполнять 2 типа запросов:

- «1  $v$   $d$   $c$ » — покрасить все вершины на расстоянии не более  $d$  от  $v$  в цвет  $c$ . Изначально все вершины имеют цвет 0.
- «2  $v$ » — вывести цвет вершины  $v$ .

### Формат входных данных

Первая строка содержит целое число  $n$  ( $1 \leq n \leq 10^5$ ) — количество вершин в дереве.

Следующие  $n - 1$  содержат тройки чисел  $u_i, v_i, w_i$  ( $1 \leq u_i, v_i \leq n, 1 \leq w_i \leq 10^4$ ).  $i$ -е ребро соединяет вершины  $u_i, v_i$  и имеет вес  $w_i$ .

В следующей строке содержится количество запросов  $q$  ( $1 \leq q \leq 10^5$ ).

Каждая из следующих  $q$  строк содержит запрос какого-то типа:

- 1  $v$   $d$   $c$  ( $1 \leq v \leq n, 0 \leq d \leq 10^9, 0 \leq c \leq 10^9$ ).
- 2  $v$  ( $1 \leq v \leq n$ ).

### Формат выходных данных

Для каждого запроса второго типа выведите ответ на него.

### Примеры

стандартный ввод	стандартный вывод
2	20
1 2 1	10
4	
1 1 1 10	
1 1 0 20	
2 1	
2 2	
5	6
1 2 30	6
1 3 50	0
3 4 70	5
3 5 60	7
8	
1 3 72 6	
2 5	
1 4 60 5	
2 3	
2 2	
1 2 144 7	
2 4	
2 5	

## Задача D. Почтовая реформа

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В Флатландии идет пора реформ. Недавно была проведена реформа дорог, так что теперь по дорогам страны из любого города можно добраться в любой другой, причем только одним способом. Также была проведена реформа волшебников, так что в каждом городе остался ровно один волшебник. Теперь же началась реформа почтовой системы.

Недавно образованное почтовое агентство «Экс-Федя» предлагает уникальную услугу — коллективную посылку. Эта услуга позволяет отправлять посылки жителям всех городов на каком-либо пути по цене обычной посылки. Удивительно, но пользоваться такой услугой стали только волшебники Флатландии, которые стали в большом количестве отправлять друг другу магические кактусы. Агентство столкнулось с непредвиденной проблемой: как известно, все волшебники живут в башнях и мало того, что не строят в них лестницы, так еще время от времени меняют их высоту. Поэтому, чтобы доставить посылку волшебнику, который живет в башне высотой  $h$ , курьеру агентства требуется иметь с собой не менее  $h$  метров веревки.

Вам поручено руководить отделом логистики — по имеющимся данным о высотах башен и об их изменениях вам нужно определять минимальную длину веревки, которую нужно выдать курьеру, который доставляет посылки между городами  $i$  и  $j$ .

### Формат входных данных

Первая строка входного файла содержит число  $n$  — количество городов в Флатландии ( $1 \leq n \leq 50\,000$ ). Во второй строке находится  $n$  положительных чисел, не превосходящих  $10^5$  — высоты башен в городах. В следующих  $n - 1$  строках содержится по два числа  $u_i$  и  $v_i$  — описание  $i$ -й дороги,  $1 \leq u_i, v_i \leq n, u_i \neq v_i$ . В следующей строке содержится число  $k$  — количество запросов ( $1 \leq k \leq 100\,000$ ). В следующих  $k$  строках содержатся описания запросов в следующем формате:

- Уведомление от волшебника из города  $i$  о том, что высота его башни стала равна  $h$ , имеет вид  $! i h, 1 \leq i \leq n, 1 \leq h \leq 10^5$ .
- Запрос от курьера о выдаче веревки для доставки посылок во все города на пути от  $i$  до  $j$  включительно имеет вид  $? i j, 1 \leq i, j \leq n$ .

### Формат выходных данных

Для каждого запроса доставки посылок выведите минимальную длину веревки, которую необходимо выдать курьеру.

## Примеры

стандартный ввод	стандартный вывод
3 1 2 3 1 3 2 3 5 ? 1 2 ! 1 5 ? 2 3 ! 3 2 ? 1 2	3 3 5
1 100 5 ! 1 1 ? 1 1 ! 1 1000 ? 1 1 ! 1 1	1 1000

## Задача Е. Найти количество

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.6 секунд  
Ограничение по памяти: 256 мегабайт

Дано подвешенное дерево из  $n$  вершин с корнем в 1. Необходимо обработать  $q$  запросов  $(v_i, h_i)$ : найти количество вершин, лежащих в поддереве  $v_i$ , расстояние до которых от  $v_i$  равно  $h_i$ . Расстояние между вершинами — минимальное количество рёбер в пути между ними.

### Формат входных данных

Первая строка содержит  $n$  ( $1 \leq n \leq 3 \cdot 10^5$ ).

Следующая строка содержит  $n - 1$  число  $p_2, \dots, p_n$  ( $1 \leq p_i \leq i$ ).  $p_i$  — отец вершины  $i$ .

Следующая строка содержит число  $q$  ( $1 \leq q \leq 3 \cdot 10^5$ ).

Следующие  $q$  строк содержат числа  $v_i, h_i$  ( $1 \leq v_i \leq n, 1 \leq h_i \leq n$ ).

### Формат выходных данных

Для каждого запроса выведите ответ.

### Пример

стандартный ввод	стандартный вывод
5	2
1 2 2 1	1
3	0
1 1	
2 0	
3 5	

## Задача F. Peterson Polyglot

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.25 секунд
Ограничение по памяти:	256 мегабайт

Петрович обожает изучать новые языки, но самое любимое его увлечение — составление своих собственных. Языком Петрович называет множество слов, а словом — последовательность маленьких букв латинского алфавита.

Каждое утро Петрович составляет свой новый язык. Хранить все языки в явном виде очень сложно, поэтому Петрович придумал веник — специальную структуру данных для хранения языка, представляющую собой подвешенное дерево, на ребрах которого написаны буквы. Перед придумыванием языка веник представляет одну вершину — корень. При добавлении нового слова в язык Петрович встает в корень веника и обрабатывает буквы слова по одной. Пусть Петрович стоит в вершине  $u$ . Если из  $u$  есть ребро, на котором написана текущая буква, он переходит по нему. Иначе же, Петрович добавляет ребро из  $u$  в новую вершину  $v$ , пишет на нем текущую букву и переходит по этому ребру. Размером веника Петрович называет количество вершин в нем.

Вечером, приходя со смены, Петрович не может понять язык, придуманный утром: он ему кажется слишком сложным. Тогда Петрович старается упростить свой язык. Упрощением языка Петрович называет удаление букв из некоторых слов языка. Формально, Петрович фиксирует некоторое целое положительное число  $p$ , берет все слова, содержащие хотя бы  $p$  букв, и выкидывает из каждого из них букву с номером  $p$ . Буквы в слове Петрович предпочитает нумеровать, начиная с 1. Петрович считает, что при упрощении языка хотя бы одно слово должно измениться, то есть в языке должно быть хотя бы слово с длиной хотя бы  $p$ . Так как Петрович стремится сделать язык, придуманный утром, как можно проще, он старается подобрать число  $p$  таким образом, чтобы минимизировать размер веника, в котором он будет хранить язык.

Петровичу надоело заниматься одним и тем же каждый вечер, поэтому он обратился за помощью к вам. Напишите программу, которая будет находить минимальный размер веника, который может получиться в результате упрощения языка, придуманного Петровичем, и число  $p$ , которое нужно выбрать, чтобы получить такой размер.

### Формат входных данных

В первой строке входного файла задано число  $n$  ( $2 \leq n \leq 3 \cdot 10^5$ ) — размер веника языка Петровича.

В следующих  $n - 1$  строках задано описание веника. В  $i$ -й из них записаны числа  $u_i, v_i$  и буква  $x_i$ , что соответствует ребру из  $u_i$  в  $v_i$ , на котором написана буква  $x_i$ .

Вершины пронумерованы числами от 1 до  $n$ . Все  $x_i$  являются маленькими буквами латинского алфавита. Вершина с номером 1 является корнем веника.

Гарантируется, что ребра описывают корректный веник, построенный по языку Петровича.

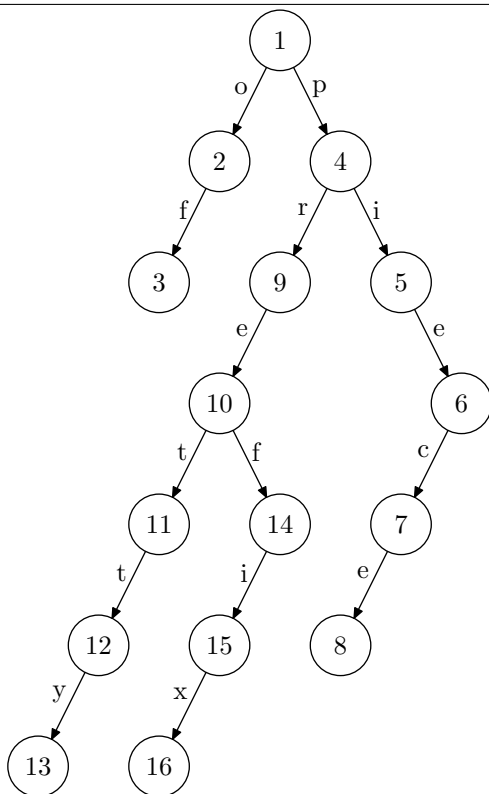
### Формат выходных данных

В первой строке выведите минимальный возможный размер веника, который может получиться в результате упрощения языка.

Во второй строке выведите число  $p$ , которое следует выбрать Петровичу для получения минимального размера. Если таких чисел  $p$  несколько, выведите минимальное из них.

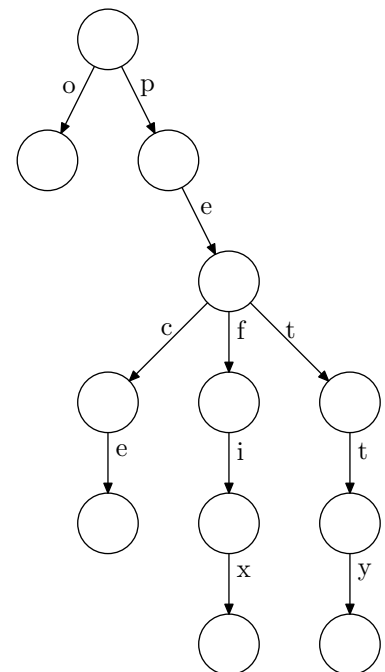
## Примеры

стандартный ввод	стандартный вывод
5 1 2 c 2 3 a 3 4 t 2 5 t	3 2
16 1 2 o 2 3 f 1 4 p 4 5 i 5 6 e 6 7 c 7 8 e 4 9 r 9 10 e 10 11 t 11 12 t 12 13 y 10 14 f 14 15 i 15 16 x	12 2



Веник для исходного языка из примера 2.

→



Веник после упрощения при  $p = 2$ .

Веник из второго примера может быть составлен из множества слов «*piece*», «*of*», «*pie*», «*pretty*», «*prefix*». После упрощения языка с  $p = 2$  получается язык из слов «*ресе*», «*о*», «*ре*», «*retty*», «*refix*». Этот язык и задаёт веник минимального возможного размера.



## Задача G. Дерево и запросы

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Задано корневое дерево, состоящее из  $n$  вершин. Каждая вершина дерева имеет определенный цвет. Будем считать, что вершины дерева пронумерованы целыми числами от 1 до  $n$ . Тогда цвет вершины  $v$  будем обозначать  $c_v$ . Корнем дерева является вершина с номером 1.

В задаче вам требуется ответить на  $m$  запросов. Каждый запрос характеризуется двумя целыми числами  $v_j, k_j$ . Ответ на запрос  $v_j, k_j$  – это количество таких цветов вершин  $x$ , что в поддереве вершины  $v_j$  содержится как минимум  $k_j$  вершин цвета  $x$ .

### Формат входных данных

В первой строке записаны два целых числа  $n$  и  $m$  ( $2 \leq n \leq 10^5$ ;  $1 \leq m \leq 10^5$ ). В следующей строке записана последовательность целых чисел  $c_1, c_2, \dots, c_n$  ( $1 \leq c_i \leq 10^5$ ). В следующих  $n - 1$  строках записаны ребра дерева. В  $i$ -ой строке записаны числа  $a_i, b_i$  ( $1 \leq a_i, b_i \leq n$ ;  $a_i \neq b_i$ ) – вершины, соединенные ребром дерева.

Далее в  $m$  строках записаны запросы. В  $j$ -той строке записаны два целых числа  $v_j, k_j$  ( $1 \leq v_j \leq n$ ;  $1 \leq k_j \leq 10^5$ ).

### Формат выходных данных

Выведите  $m$  целых чисел – ответы на запросы в порядке появления запросов во входных данных.

### Пример

стандартный ввод	стандартный вывод
8 5	2
1 2 2 3 3 2 3 3	2
1 2	1
1 5	0
2 3	1
2 4	
5 6	
5 7	
5 8	
1 2	
1 3	
1 4	
2 3	
5 3	

## Задача Н. Найти количество

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дано подвешенное дерево из  $n$  вершин с корнем в 1. Необходимо обработать  $q$  запросов  $(v_i, h_i)$ : найти количество вершин, лежащих в поддереве  $v_i$ , расстояние до которых от  $v_i$  равно  $h_i$ . Расстояние между вершинами — минимальное количество рёбер в пути между ними.

### Формат входных данных

Первая строка содержит  $n$  ( $1 \leq n \leq 3 \cdot 10^5$ ).

Следующая строка содержит  $n - 1$  число  $p_2, \dots, p_n$  ( $1 \leq p_i \leq i$ ).  $p_i$  — отец вершины  $i$ .

Следующая строка содержит число  $q$  ( $1 \leq q \leq 3 \cdot 10^5$ ).

Следующие  $q$  строк содержат числа  $v_i, h_i$  ( $1 \leq v_i \leq n, 1 \leq h_i \leq n$ ).

### Формат выходных данных

Для каждого запроса выведите ответ.

### Пример

стандартный ввод	стандартный вывод
5	2
1 2 2 1	1
3	0
1 1	
2 0	
3 5	

## Задача I. Столицы

Имя входного файла: `capitals.in`  
Имя выходного файла: `capitals.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

В стране Триландии близятся выборы новых столиц. Столицы в Триландии необычные, поскольку ими являются одновременно сразу три различных города. Такая идея размещения столиц основана на исследованиях эффективности управления страной, выполненных ведущими экономистами Триландии.

Всего в Триландии  $n$  городов, из которых некоторые пары городов соединены дорогами, и по каждой из них можно проехать в обе стороны. Время проезда по каждой дороге в одну сторону равно одному часу. При этом все города соединены дорогами таким образом, что из каждого города можно добраться в любой другой, причем это можно сделать единственным способом, если по каждой дороге проезжать не более одного раза и только в одну сторону.

Как показали результаты проведенных триландскими экономистами исследований, управление страной будет наиболее эффективным, если три столицы будут выбраны так, что время кратчайшего пути между каждой парой столиц составит ровно  $d$  часов. Перед проведением выборов необходимо знать, сколько существует различных троек городов, удовлетворяющих описанным выше свойствам. Две тройки городов считаются различными, если в первой тройке есть хотя бы один город, которого нет во второй тройке, и наоборот.

Требуется написать программу, которая по количеству городов в Триландии и описанию дорог находит количество троек городов, которые могут быть столицами.

### Формат входных данных

Первая строка входного файла содержит два разделенных пробелом целых числа: количество городов в Триландии  $n$  и требуемое время в пути между столицами  $d$  ( $3 \leq n \leq 10^5$ ,  $1 \leq d < n$ ). Каждая из последующих  $(n - 1)$  строк содержит описание одной дороги: пару разделенных пробелом различных целых чисел  $a_i$  и  $b_i$  — номера городов, которые соединены двусторонней дорогой ( $1 \leq a_i \leq n$ ,  $1 \leq b_i \leq n$ ,  $a_i \neq b_i$ ). Каждая пара городов соединена не более чем одной дорогой.

### Формат выходных данных

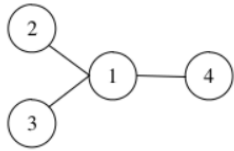
Выходной файл должен содержать одно целое число — количество подходящих троек городов, которые могут быть выбраны столицами. В случае, если нужных троек городов не окажется, выходной файл должен содержать ноль.

### Примеры

<code>capitals.in</code>	<code>capitals.out</code>
4 2 1 2 1 3 1 4	1
7 2 1 2 1 3 1 4 5 1 5 6 5 7	5

### Замечание

В первом примере существует единственный способ выбрать три столицы: города под номерами 2, 3 и 4.



Во втором примере существует четыре варианта выбора трёх столиц из четверки городов: 2, 3, 4 и 5. Можно также выбрать столицами города с номерами 1, 6 и 7.

