

## Задача А. Командир Ciel

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

Лиса Ciel становится командиром Древоземелья. В Древоземелье, как это следует из названия, есть  $n$  городов, соединенных  $n - 1$  ненаправленными дорогами, а между любыми двумя городами существует путь по дорогам Древоземелья.

Лиса Ciel должна назначить каждому городу офицера. У каждого офицера есть ранг — буква от «А» до «Z». Таким образом, имеется 26 различных рангов, самый высокий — «А», самый низкий — «Z».

У Ciel имеется достаточно офицеров каждого ранга. Но не все так просто, должно быть выполнено особое условие: если  $x, y$  — два различных города и у их офицеров одинаковые ранги, то на простом пути между  $x$  и  $y$  должен быть город  $z$ , имеющий офицера более высокого ранга. Таким образом, общение между офицерами одного ранга будет гарантированно проходить под присмотром офицера более высокого ранга.

Помогите Ciel составить подходящий план назначения офицеров городам. Если это невозможно, выведите «Impossible!».

### Формат входных данных

В первой строке записано целое число  $n$  ( $2 \leq n < 10^5$ ) — количество городов в Древоземелье.

В каждой из следующих  $n - 1$  строк записано два целых числа  $a$  и  $b$  ( $1 \leq a, b \leq n, a \neq b$ ) — это значит, что существует дорога между городами  $a$  и  $b$ . Считайте, что города пронумерованы от 1 до  $n$  некоторым образом.

Гарантируется, что заданный граф будет деревом.

### Формат выходных данных

Если подходящий план существует, выведите  $n$  символов, разделенных пробелами —  $i$ -ый символ обозначает ранг офицера в городе  $i$ . В противном случае, выведите «Impossible!».

### Примеры

стандартный ввод	стандартный вывод
4 1 2 1 3 1 4	A B B B
10 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10	D C B A D C B D C D

## Задача В. Найти ближайшую

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 3 секунды  
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: 3 секунды  
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. Сигнализация

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Подземный бункер состоит из  $n$  комнат, соединённых  $n - 1$  коридорами. Каждый коридор соединяет две различные комнаты и имеет определённую длину. Бункер устроен таким образом, что из любой комнаты  $i$  можно прийти в любую другую комнату  $j$ . Заметим, что существует единственный такой путь, не проходящий по одному и тому же коридору дважды. Сумма длин коридоров, составляющих этот путь, называется расстоянием между комнатами  $i$  и  $j$  и обозначается  $\rho(i, j)$ .

Каждая комната бункера оборудована звуковой сигнализацией, состоящей из сирены и датчика звука, который её включает. Сирена, включённая в комнате  $i$ , активирует датчик звука в каждой комнате, расстояние до которой не превосходит расстояние  $d_i$ , определяемое мощностью этой сирены. Другими словами, включение сирены в комнате  $i$  автоматически включает сирену во всех комнатах  $j$ , таких что  $\rho(i, j) \leq d_i$ . Эта сирена, в свою очередь, может вызвать автоматическое включение других сирен и так далее.

В случае возникновения чрезвычайной ситуации некоторые сирены необходимо включить вручную, после чего звук от них автоматически включит сирены в других комнатах. Правила безопасности предписывают выбор такого набора сирен для ручного включения, который в конце концов приведёт к автоматическому включению сирен во всех комнатах.

Требуется написать программу, которая определяет минимальное количество сирен в наборе, удовлетворяющем правилам безопасности.

### Формат входных данных

Первая строка входных данных содержит единственное число  $n$  ( $1 \leq n \leq 100\,000$ ) — количество комнат.

Вторая строка содержит последовательность из  $n$  целых чисел  $d_i$ ,  $i$ -е из них равно максимальному расстоянию, на котором расположенная в комнате  $i$  сирена активирует датчики ( $0 \leq d_i \leq 10^9$ ).

Последующие  $n - 1$  строк описывают коридоры бункера. В  $i$ -й из них находятся три целых числа:  $u_i, v_i, l_i$ , где  $u_i, v_i$  — номера различных комнат, соединённых коридором  $i$ , а  $l_i$  — длина этого коридора ( $1 \leq u_i, v_i \leq n; 1 \leq l_i \leq 10^9$ ).

### Формат выходных данных

Выходные данные должны состоять из единственного числа — минимального количества сирен, которые необходимо включить вручную.

### Пример

стандартный ввод	стандартный вывод
10 1 2 2 2 6 3 4 5 4 3 1 2 5 2 3 1 2 4 5 4 5 2 4 6 4 4 7 3 1 8 1 8 9 5 8 10 4	3

### Замечание

В тесте из примера сирена в комнате 4 включает сирену в комнате 5, которая, в свою очередь, включает сирены в комнатах 6 и 7. Сирена в комнате 2 включает сирену в комнате 3. Сирена в комнате 8 включает сирены в комнатах 1, 9 и 10.

## Задача Е. 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	

### Замечание

А за сколько работает решение? Пишите @qoo2p5, если хотите проверить себя. :)

## Задача F. Peterson Polyglot

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
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.

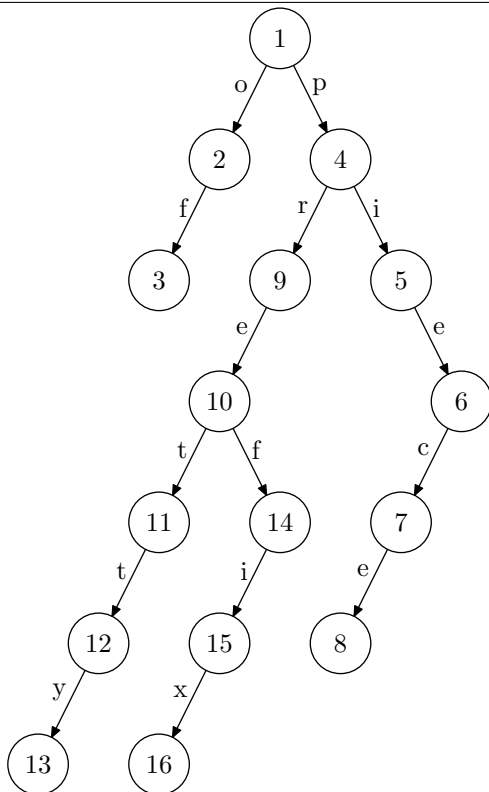
### Формат выходных данных

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.

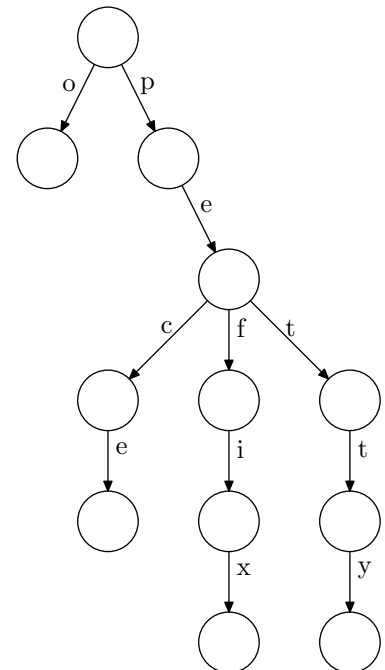
Примеры

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 «pесе», «о», «ре», «retty», «pefix». Этот язык и задаёт веник минимального возможного размера.