

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 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 С. 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 D. 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 E. В бухгалтерии опять всё перепутали

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

Лула и Пула пошли получать зарплату. Но в бухгалтерии опять всё перепутали. Лула получил зарплату за Пулу, а Пула . . .

Пула не хочет получать за Луну и хочет доказать бухгалтерии, что она не права.

Пула работает в крупной компании «MST Inc.», занимающейся информационным сопровождением «Всеберляндской олимпиады школьников по информатике». В компании «MST Inc.» работает n сотрудников, причём у каждого из них, кроме самой «MST», есть ровно один непосредственный начальник и несколько (возможно ноль) непосредственных подчинённых.

Всеми начальниками сотрудника компании «MST Inc.» называется множество, состоящее из его непосредственного начальника и множества начальников его непосредственного начальника. Известно, что у каждого сотрудника кроме самой «MST», «MST» входит в множество начальников этого сотрудника.

Множеством подчинённых у сотрудника называется множество, состоящее из него самого и множеств подчинённых у всех непосредственных подчинённых данного сотрудника. В частности, все сотрудники входят в множество подчинённых у «MST».

Каждый месяц каждому сотруднику начисляется зарплата, причём немаленькая, ведь иначе ни один сотрудник не согласился бы работать с «MST». Известно, что в нулевой месяц работы организации, каждому сотруднику заплатили по c_i бурлей. В качестве поощрения сотрудников «MST» придумала следующее правило: В каждый из следующих m месяцев берётся сотрудник с номером a_i и берётся число s_i — сумма зарплат всех сотрудников во множестве его начальников и подчинённых (включая его самого). Если это число оказывалось слишком большим, s_i берётся по модулю $10^9 + 7$. После этого берётся сотрудник с номером b_i , и к зарплате всех сотрудников, входящих во множество его начальников и подчинённых (включая его самого) прибавляется число s_i . С учётом этого изменения платится зарплата в i -й месяц и пересчитывается зарплата в следующие месяцы.

Вернёмся к Пуле. Пула хочет показать бухгалтерии компании «MST Inc.» что она всё перепутала, а для этого ему надо узнать, сколько же ему должны были заплатить в каждый из месяцев с нулевого по m -й. К сожалению, в гениальной системе поощрения, разработанной «MST», не может разобраться никто. Поэтому эту задачу поручили вам.

Формат входных данных

В первой строке входных данных даны 2 числа n и m ($1 \leq n, m \leq 10^5$) — число сотрудников компании «MST Inc.» и последний день, когда выплачивалась зарплата Пуле.

Во второй строке записано $n - 1$ число. i -е из них — номер непосредственного начальника сотрудника номер i (i принимает значения от 1 до $n - 1$). При этом «MST» имеет номер 0 и не имеет непосредственного начальника. Пула имеет номер $n - 1$.

В третьей строке записано n чисел c_i ($1 \leq c_i \leq 10^9$) — зарплата i -го сотрудника в нулевой день.

В каждой из следующих m строк записано по 2 числа a_i и b_i ($0 \leq a_i, b_i \leq n - 1$) — номер человека, на основе которого происходит поощрение и номер человека, к подчинённым и начальникам которого поощрение применяется (более подробно описано в условии).

Формат выходных данных

В единственной строке выведите $m + 1$ число — зарплату Пулы в каждый из дней с 0-го по m -й. Напоминаем, что Пула имеет номер $n - 1$. Обратите внимание, что зарплата **не считается** по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
3 3 0 0 1 1 1 0 0 2 1 1 2	1 4 4 28
4 3 0 1 1 0 1 0 0 0 1 1 3 2 3	0 1 6 20

Замечание

Пояснение к первому примеру:

В первый день к зарплате каждого сотрудника прибавилось 3 бурля и зарплаты стали соответственно 4, 4, 4.

Во второй день к зарплате сотрудников с номерами 0, 1 прибавилось по 8 бурлей и зарплаты стали соответственно 12, 12, 4.

Во третий день к зарплате сотрудников с номерами 0, 2 прибавилось по 24 бурля и зарплаты стали соответственно 36, 12, 28.

Задача F. Почтовая реформа

Имя входного файла:	mail.in
Имя выходного файла:	mail.out
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 ($1 \leq n \leq 50000$). 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 ($1 \leq k \leq 100000$). 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

mail.in	mail.out
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 G. Гоша и праздники

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

Как известно, жители планеты Иннополис — очень педантичные люди. И даже когда дело касается праздников, они всегда хотят быть уверенными в том, что все пройдёт как по маслу. Так, расписание празднований всех событий на этой планете составлено почти на три миллиона лет вперёд! Гоша — большой любитель праздников. Он решил прилететь в какой-то из городов планеты Иннополис и посетить как можно больше праздников.

На планете Иннополис n городов, соединённых $n - 1$ двунаправленными дорогами так, что из любого города планеты можно добраться до любого другого, возможно, посещая другие города. Каждое событие на Иннополисе характеризуется номером города c_i , в котором оно будет отпраздновано, и номером дня d_i , в который его будут праздновать.

Гоша настолько везучий человек, что день его прибытия на планету имеет номер 0 в календаре планеты Иннополис, причём исходно он может прилететь в любой город планеты. Гоша решил узнать, какое максимальное количество праздников он может посетить на этой планете. Для этого он обратился за помощью к вам.

Формат входных данных

В первой строке входного файла задано одно число n ($n \geq 1$) — количество городов Иннополиса.

В следующих $n - 1$ строках заданы описания дорог, каждая дорога задается числами a_i , b_i и l_i ($1 \leq a_i, b_i \leq n$; $l_i \geq 1$) — номера городов, которые соединяет дорога и число дней, необходимых на ее преодоление.

В следующей строке задано число m ($m \geq 1$) — число праздников на планете.

В следующих m строках заданы пары чисел c_i и d_i ($1 \leq c_i \leq n$; $d_i \geq 1$) — номер города и номер дня, в который пройдёт i -й праздник.

Ограничения: $n \leq 2 \cdot 10^5$, $m \leq 2 \cdot 10^5$, $l_i \leq 10^9$, $d_i \leq 10^9$.

Формат выходных данных

В единственной строке выходного файла выведите одно число — максимальное количество праздников, которое может посетить Гоша.

Примеры

стандартный ввод	стандартный вывод
4 1 2 1 2 3 1 2 4 3 4 1 3 2 4 3 1 4 5	3
11 2 1 2 3 2 5 4 1 5 5 2 4 6 5 1 7 1 2 8 3 4 9 6 2 10 7 2 11 2 2 9 1 67 1 34 11 16 5 97 4 70 2 20 2 61 2 26 2 70	8
10 2 1 1 3 2 4 4 2 4 5 3 2 6 4 5 7 5 4 8 3 1 9 6 2 10 7 5 9 7 34 10 82 2 48 3 66 8 98 2 66 3 3 8 59 5 22	8

Задача N. Ближайший лист

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

Определим эйлеров обход дерева (связного неориентированного графа без циклов) следующим образом: рассмотрим рекурсивный алгоритм поиска в глубину, который обходит вершины дерева и нумерует вершины в том порядке, в котором их посещает, при этом учитывается только первое посещение каждой вершины. Данная функция стартует из вершины с номером 1, а затем рекурсивно вызывается от всех вершин, которые соединены ребром с текущей и ещё не посещены, в порядке возрастания номеров вершин. Формально данную функцию можно описать так:

```
next_id = 1
id = массив длины n, заполненный -1
visited = массив длины n, заполненный false

function dfs(v):
    visited[v] = true
    id[v] = next_id
    next_id += 1
    for to по соседям v в порядке возрастания:
        if not visited[to]:
            dfs(to)
```

Дано взвешенное дерево, вершины которого пронумеровали в порядке эйлерова обхода целыми числами от 1 до n при помощи алгоритма, описанного выше.

Назовём листом вершину дерева, соединённую ребром ровно с одной другой вершиной. В данном вам дереве вершина 1 не является листом. Расстоянием между двумя вершинами дерева назовём сумму весов рёбер на единственном простом пути между ними.

Требуется ответить на q запросов следующего вида: по заданным числам v , l и r сообщить кратчайшее расстояние от вершины v до одного из листов дерева, имеющего номер от l до r (включительно).

Формат входных данных

В первой строке даны два целых числа n и q ($3 \leq n \leq 500\,000, 1 \leq q \leq 500\,000$) — количество вершин в дереве и количество запросов соответственно.

Следующие $n - 1$ строк задают рёбра дерева: $(i - 1)$ -я строка содержит два целых числа p_i и w_i ($1 \leq p_i < i, 1 \leq w_i \leq 10^9$), обозначающие ребро между вершинами p_i и i с весом w_i .

Гарантируется, что заданные рёбра задают дерево, вершины которого пронумерованы в порядке эйлерова обхода, и что вершина с номером 1 не является листом.

Следующие q строк содержат описания запросов. Каждая из них содержит три целых числа v_i, l_i, r_i ($1 \leq v_i \leq n, 1 \leq l_i \leq r_i \leq n$), обозначающие параметры запроса, описанные в условии. Гарантируется, что существует хотя бы один лист с номером x такой, что $l_i \leq x \leq r_i$.

Формат выходных данных

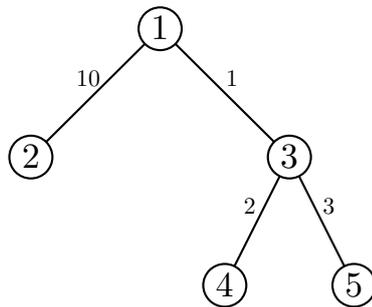
Выведите q чисел — ответы на запросы в порядке, в котором они заданы во входных данных.

Примеры

стандартный ввод	стандартный вывод
5 3 1 10 1 1 3 2 3 3 1 1 5 5 4 5 4 1 2	3 0 13
5 3 1 1000000000 2 1000000000 1 1000000000 1 1000000000 3 4 5 2 1 5 2 4 5	3000000000 1000000000 2000000000
11 8 1 7 2 1 1 20 1 2 5 6 6 2 6 3 5 1 9 10 9 11 5 1 11 1 1 4 9 4 8 6 1 4 9 7 11 9 10 11 8 1 11 11 4 5	8 8 9 16 9 10 0 34

Замечание

В первом примере дерево выглядит так:



В первом запросе ближайший к вершине 1 лист имеет номер 4. Расстояние до него равно 3. Во втором запросе ближайшим к вершине 5 листом является вершина с номером 5, расстояние до которой равно 0. В третьем примере ближайшим к вершине 4 листом является вершина с номером 4, однако она не попадает в отрезок вершин $[1, 2]$ запроса. Единственным листом с номером, попадающим в отрезок $[1, 2]$ является вершина с номером 2, расстояние до которой от вершины 4 равно 13.

Задача I. 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.

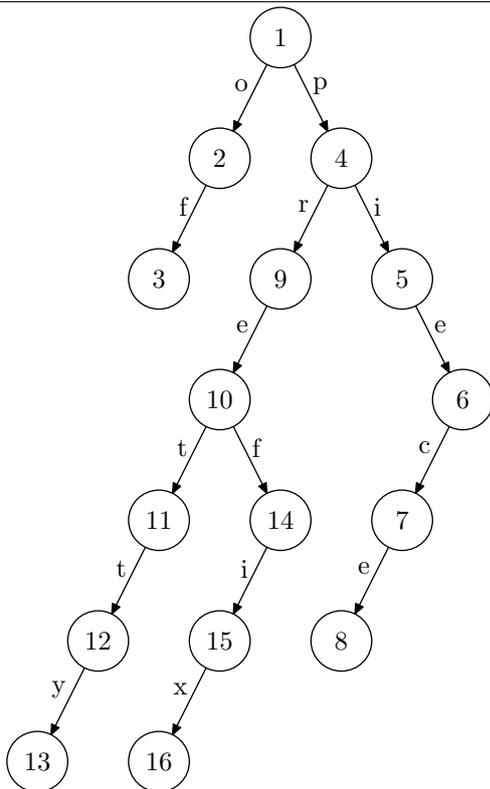
Формат выходных данных

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.

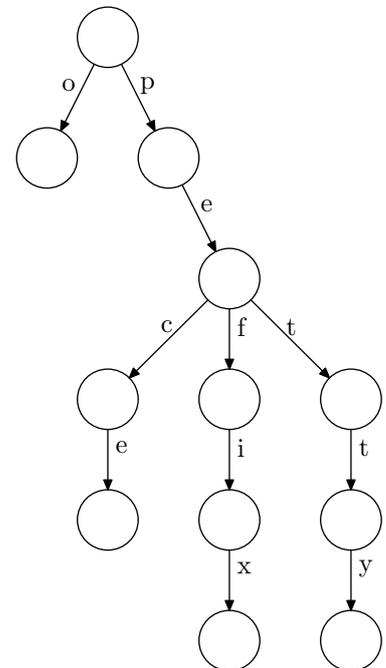
Примеры

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 «*ресе*», «*о*», «*ре*», «*retty*», «*реfix*». Этот язык и задаёт веник минимального возможного размера.

Задача J. Стратегически важные пути

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

В королевстве Берляндия есть n городов, соединённых $n - 1$ двусторонней дорогой, такие что между каждой парой городов есть путь. У каждого города есть его идентификатор — буква английского алфавита от **a** до **t**.

Недавно король королевства выпустил указ, гласящий что путь в королевстве называется стратегически важным, если в нём найдётся не более одного идентификатора, такого что городов на этом пути с этим идентификатором нечётное число.

Теперь жителям каждого города интересно количество стратегически важных путей, проходящих через данный город. Помогите жителям каждого города выяснить это.

Формат входных данных

В первой строке содержится целое число n ($2 \leq n \leq 2 \cdot 10^5$) — количество городов.

В следующих $n - 1$ строках содержатся пары чисел u и v ($1 \leq u, v \leq n, u \neq v$), обозначающие дорогу между городами u и v . Гарантируется, что между каждой парой городов есть путь (возможно не стратегически важный).

В следующей строке содержится строка, состоящая из n символов английского алфавита от **a** до **t**, где i -й ($1 \leq i \leq n$) символ обозначает идентификатор i -го города.

Формат выходных данных

Выведите n чисел, где i -е число обозначает количество стратегически важных путей, проходящих через город i .

Учтите, что путь из города u в город v и путь из города v в города u считаются одинаковыми, такие пути должны учитываться ровно один раз.

Система оценки

Решения, правильно работающие при $n \leq 500$ будут оцениваться в 28 баллов.

Дополнительные 29 баллов получают решения, правильно работающие при $n \leq 5000$.

Пример

стандартный ввод	стандартный вывод
5	1 3 4 3 3
1 2	
2 3	
3 4	
3 5	
abcbb	

Замечание

В первом тестовом примере следующие пути являются стратегически важным:

2 – 3 – 4

2 – 3 – 5

4 – 3 – 5

Кроме того, все пути, состоящие из одной вершины, являются стратегически важными. А следующие пути стратегически важными не являются:

1 – 2 – 3

1 – 2 – 3 – 4

1 – 2 – 3 – 5