

## 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: 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. 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 E. 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 F. 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



## 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  $n - 1$  двусторонними 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 Н. Нет монет

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Древляндия — страна из  $n$  городов, соединённых  $n - 1$  дорогой так, что любые два города достижимы друг из друга по дорогам (возможно, не напрямую). Есть  $m$  запросов на транспортировку товаров,  $i$ -й запрос представляет собой доставку между городами  $u_i, v_i$  и даёт  $p_i$  монет.

Вы — маленькая лошадка, у которой пока что нет монет. Но вы очень хотите их заработать. Для этого вы хотите выбрать два города  $x$  и  $y$  и осуществить все транспортировки, у которых конечные города лежат на кратчайшем пути между  $x$  и  $y$ . Конечно, вы хотите выбрать  $x$  и  $y$  так, чтобы заработать как можно больше монет.

### Формат входных данных

Первая строка содержит целое число  $n$  ( $2 \leq n \leq 10^5$ ).

Следующие  $n - 1$  строк содержат описание дорог.  $i$ -я из этих строк содержит два целых числа  $a_i, b_i$  ( $1 \leq a_i, b_i \leq n$ ) — города, которые соединяет  $i$ -я дорога.

Вторая строка содержит целое число  $m$  ( $0 \leq m \leq 10^5$ ).

Следующие  $m$  строк содержат описание запросов на транспортировку.  $i$ -я из этих строк содержит три целых числа  $u_i, v_i, p_i$  ( $1 \leq u_i, v_i \leq n, 1 \leq p_i \leq 10^3$ ).

### Формат выходных данных

Выведите единственное число — максимальное количество монет, которое вы можете получить.

### Пример

стандартный ввод	стандартный вывод
6 1 2 2 3 2 4 5 4 6 4 4 1 4 10 2 5 20 6 3 15 2 1 1	31

### Замечание

В примере оптимально выбрать  $x = 1$  и  $y = 5$ .

## Задача I. [A] 5. Запоминающийся путь

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 256 мегабайт

Родители отправляют ребенка в школу на автобусе. Всего в городе есть  $n$  остановок, соединённых  $n - 1$  проездом так, что от любой остановки можно добраться до любой другой. Каждая остановка покрашена одним из 26 возможных цветов, обозначенных буквами латинского алфавита от «a» до «z».

Родители хотят выбрать две остановки, на одной из них построить свой дом, а на другой построить школу для ребёнка. При этом они хотят выбрать такие две остановки для школы и дома, что последовательность цветов остановок на кратчайшем пути от дома до школы должна совпадать с последовательностью цветов остановок на кратчайшем пути от школы до дома. Найдите длину самого большого пути от школы до дома, соблюдая эти условия. Дом и школа могут находиться на любых остановках.

### Формат входных данных

Первая строка содержит одно целое число  $n$  ( $1 \leq n \leq 50\,000$ ) — количество остановок в городе.

Вторая строка содержит одну строку  $s$  из  $n$  строчных латинских букв — цвета остановок.

В  $i$ -й из следующих  $n - 1$  строк задано два целых числа  $u_i$  и  $v_i$  ( $1 \leq u_i, v_i \leq n$ ), обозначающих очередную дорогу на схеме городского транспорта.

### Формат выходных данных

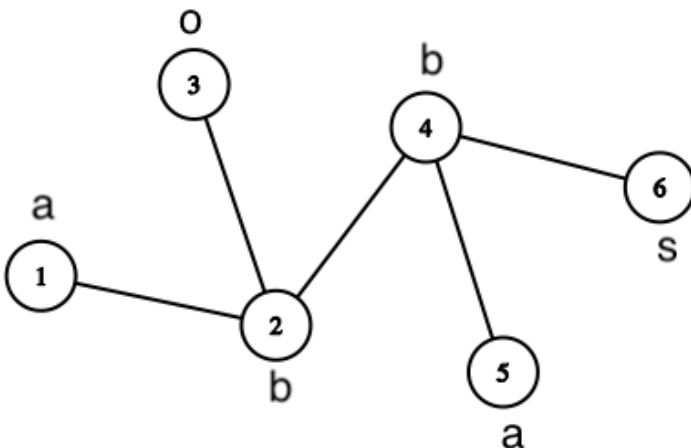
Выведите единственное число — максимальную длину пути между двумя остановками, такую что последовательность цветов остановок на прямом и обратном движении по пути совпадают.

### Пример

стандартный ввод	стандартный вывод
6 abobas 1 2 2 3 5 4 6 4 2 4	4

### Замечание

Последовательность цветов на пути 1 – 2 – 4 – 5 совпадает с последовательностью цветов на пути 5 – 4 – 2 – 1. Это самый большой путь, удовлетворяющий условию, а его длина равна 4.



## Задача J. Новые кампусы!

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	1024 мегабайта

Недавно вы стали ректором одного из университетов и решили открыть в нем новую программу! На ней вы будете учить студентов спортивному программированию. Поэтому у них будут два типа занятий: спорт (чтобы развить силу рук) и программирование. Основным достоинством этой программы будет обучение в двух кампусах одновременно — по четным дням студенты будут ездить в первый кампус, а по нечетным — во второй.

Оба кампуса вашего университета устроены очень необычно: в каждом из них есть по  $n$  аудиторий, пронумерованных от 1 до  $n$ , и по  $n - 1$  переходу между ними, при этом из любой аудитории можно добраться в любую другую по переходам.

Однако вы обнаружили, что студентам сложно ориентироваться сразу в двух кампусах, и решили упростить им жизнь. Вы решили выбрать два номера аудиторий  $u$  и  $v$  ( $u \neq v$ ): в аудитории с номером  $u$  студенты будут заниматься спортом, а в аудитории с номером  $v$  — программированием. Обратите внимание, что  $u$  и  $v$  выбираются одинаковыми для обоих кампусов.

Так как вы хотите, чтобы студенты тратили меньше времени на перемещение между аудиториями, вам нужно минимизировать суммарное расстояние, которое потребуется преодолеть студентам между выбранными аудиториями в каждом из кампусов. Более формально, вам нужно найти такие номера  $u, v$ , что  $d_1(u, v) + d_2(u, v)$  минимально, где  $d_1(u, v)$  — это расстояние между аудиториями  $u$  и  $v$  в первом кампусе, а  $d_2(u, v)$  — во втором. Расстоянием между аудиториями называется минимальное число переходов, через которые нужно пройти, чтобы добраться из одной аудитории в другую.

В обоих кампусах есть вход, и он ведет в аудиторию 1. Для всех остальных аудиторий разработан план эвакуации. В первом кампусе для  $i$ -й аудитории  $p_i$  равно номеру следующей аудитории на пути из  $i$ -й аудитории в первую. Во втором кампусе для  $i$ -й аудитории  $q_i$  равно номеру следующей аудитории на пути из  $i$ -й аудитории в первую.

### Формат входных данных

В первой строке входных данных содержится одно целое число  $n$  ( $2 \leq n \leq 10^6$ ) — количество аудиторий.

В следующей строке находятся  $n - 1$  целых чисел  $p_2, p_3, p_4, \dots, p_n$  ( $1 \leq p_i \leq n$ ), где  $p_i$  — это следующая (кроме  $i$ ) аудитория на пути от  $i$ -й до первой в первом кампусе.

В следующей строке находятся  $n - 1$  целых чисел  $q_2, q_3, q_4, \dots, q_n$  ( $1 \leq q_i \leq n$ ), где  $q_i$  — это следующая (кроме  $i$ ) аудитория на пути от  $i$ -й до первой во втором кампусе.

### Формат выходных данных

В первой строке выведите минимальную величину  $d_1(u, v) + d_2(u, v)$ .

Во второй строке выведите любую пару вершин  $u, v$ , таких что  $d_1(u, v) + d_2(u, v)$  минимально.

## Примеры

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

## Замечание

В первом примере в первом кампусе есть переходы между аудиториями 3 и 2, 1 и 3, 2 и 4, 3 и 5. Во втором кампусе есть переходы между аудиториями 5 и 2, 4 и 3, 1 и 4, 3 и 5.

## Система оценки

Тесты к этой задаче состоят из 12 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Доп. ограничения	Необх. группы	Комментарий
		$n$		
0	0	–	–	Тесты из условия.
1	12	$n \leq 500$	0	
2	11	$n \leq 5000$	0, 1	
3	8	$n \leq 50\,000$	0, 1, 2	
4	11	$n \leq 100\,000$	–	В первом кампусе существует аудитория, соединенная прямыми переходами со всеми остальными аудиториями
5	12	$n \leq 100\,000$	–	В обоих кампусах для каждой аудитории существует не более двух переходов в соседние аудитории
6	10	$n \leq 100\,000$	5	В первом кампусе для каждой аудитории существует не более двух переходов в соседние аудитории
7	9	$n \leq 100\,000$	0 – 6	
8	10	$n \leq 200\,000$	0 – 7	
9	11	$n \leq 300\,000$	0 – 8	<b>Offline-проверка.</b>
10	3	$n \leq 500\,000$	0 – 9	<b>Offline-проверка.</b>
11	2	$n \leq 750\,000$	0 – 10	<b>Offline-проверка.</b>
12	1	–	0 – 11	<b>Offline-проверка.</b>

## Задача К. Yet Another Tree Problem

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	512 мегабайт

Дано дерево на  $n$  вершинах (дерево — это неориентированный связный граф без циклов). Каждое ребро имеет вес — целое число. Некоторые вершины дерева помечены. Вам нужно реализовать программу, выполняющую следующие операции с этим деревом:

1. Пометить вершину. Гарантируется, что перед этой операцией она была не помечена.
2. Сделать вершину не помеченной. Гарантируется, что до этого она была помечена.
3. Изменить вес ребра.

До выполнения всех операций и после каждой операции ваша программа должна вывести максимальное число  $x$  такое, что существует простой путь с **хотя бы двумя** помеченными вершинами на нем и суммарным весом  $x$ . Если в дереве нет двух помеченных вершин, выведите строку «BAD».

### Формат входных данных

В первой строке заданы два целых числа  $n$  и  $q$  ( $1 \leq n, q \leq 150\,000$ ), число вершин и число операций соответственно.

В следующей строке находятся  $n$  чисел, означающих, помечена ли вершина: 1 означает, что вершина помечена, 0 — не помечена.

В следующих  $n - 1$  строках задаются ребра дерева. Для всех  $i$  от 2 до  $n$  записаны два числа  $p_i$  ( $1 \leq p_i < i$ ) и  $w_i$  ( $-10^9 \leq w_i \leq 10^9$ ), которые означают, что в дереве есть ребро между вершинами  $i$  и  $p_i$  веса  $w_i$ .

В следующих  $q$  строках задаются операции. Каждая операция имеет одну из следующих форм:

- 1  $x$  ( $1 \leq x \leq n$ ): сделать вершину  $x$  помеченной,
- 2  $x$  ( $1 \leq x \leq n$ ): сделать вершину  $x$  не помеченной,
- 3  $x w$  ( $2 \leq x \leq n, -10^9 \leq w \leq 10^9$ ): сделать вес ребра между вершинами  $p_x$  и  $x$  равным  $w$ .

### Формат выходных данных

Выведите  $q + 1$  строку. В первой строке выведите ответ до выполнения операций. В каждой следующей строке выведите ответ после выполнения очередной операции.

### Система оценки

Обозначим за  $D$  максимальное число ребер, выходящих из одной вершины, а за  $H$  — максимальное число вершин на простом пути от вершины 1 до какой-то другой вершины.

Подзадача	Баллы	Ограничения		
		$n$	$q$	Additional
1	3	$n \leq 20$	$q \leq 20$	—
2	3	$n \leq 500$	$q \leq 500$	—
3	3	$n \leq 100$	$q \leq 5000$	—
4	5	$n \leq 5000$	$q \leq 5000$	—
5	7	$n \leq 10^5$	$q \leq 10^5$	$D \leq 30, H \leq 20$
6	20	$n \leq 10^5$	$q \leq 10^5$	$H \leq 20$
7	10	$n \leq 10^5$	$q \leq 10^5$	$p_v = v - 1$ для всех $v$
8	17	$n \leq 10^5$	$q \leq 10^5$	$w_i \leq 0, w \leq 0$ для всех операций
9	22	$n \leq 50\,000$	$q \leq 50\,000$	—
10	5	$n \leq 10^5$	$q \leq 10^5$	—
11	5	$n \leq 150\,000$	$q \leq 150\,000$	—



### Пример

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

### Замечание

Иллюстрации к примеру. Жирным выделен путь веса  $x$  с хотя бы двумя помеченными вершинами.

