

Задача А. Камни HARD

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Гора состоит из площадок, соединенных между собой узкими проходами. На каждой площадке лежит какое-то количество камней. У разных площадок разный рейтинг, зависящий от высоты площадки — чем выше площадка, тем больше её рейтинг среди камней. Очевидно, что на площадках с более высоким рейтингом лежит большее количество камней.

В один солнечный день главный камень одной из площадок обратил внимание, что вокруг стало очень тесно. Но камни не могут катиться вверх, а на нижние площадки они катиться не хотят из-за плохого рейтинга. Поэтому главный камень пошел на хитрость. Он выбрал M камней и сказал каждому из них, что где-то внизу появилась замечательная площадка с самым высоким рейтингом на горе. Для полной правдоподобности он составил карту площадок, на которые можно скатиться с его площадки. Всего (включая переполненную площадку) оказалось N площадок. Поскольку главный камень ленивый, он не стал рисовать все пути между площадками, а просто нарисовал минимальное количество проходов между площадками, чтобы из его площадки можно было добраться в любую, находящуюся ниже. Он показал каждому из M камней эту карту и объяснил, где находится замечательная площадка (чтобы камень запомнил путь). Причем, чтобы главные камни нижних площадок не имели к нему претензий из-за такого нашествия, некоторым камням он показал на другие площадки, чем остальным. Чтобы не создать лавину, он сказал камням катиться с небольшим интервалом между собой. Немного подумав, он решил, что можно пускать камни парами, чтобы пространство быстрее освободилось. Камни ему поверили и стали собираться в дорогу.

Камни не любят скучать, поэтому каждая пара решила посчитать, сколько времени они будут катиться вместе, и, может быть, перестроить пары, чтобы увеличить это время. На то, чтобы скатиться по проходу с одной площадки в следующую, не заходя по пути на другие площадки, камень тратит одну минуту. Но складывать числа камни не умеют, в этом им требуется ваша помощь.

Формат входных данных

В первой строке входного файла записаны два целых числа — N и K , где $1 \leq N \leq 500\,000$, $K = M/2$, $0 \leq M \leq 1000\,000$.

В следующих $(N - 1)$ строках находятся пары чисел i и j , означающие, что из i -й площадки можно напрямую скатиться на j -ю. Площадки занумерованы числами от 1 до N , верхняя площадка, откуда катятся камни, может иметь любой номер. Записанных пар номеров площадок достаточно, чтобы определить путь из самой верхней площадки до любой другой.

Дальше идут K строк, на каждой строке через пробел записана пара чисел a и b ($1 \leq a, b \leq N$), обозначающая пункты назначения для очередной пары камней: первый камень катится на площадку a , а второй — на площадку b .

Формат выходных данных

В выходной файл нужно вывести K строк, на каждой строке должно находиться по три целых числа a , b и c , записанных через пробел. Эти числа означают, что пара камней, направляющихся на площадки a и b , будет катиться вместе c минут. Тройки чисел нужно выдавать в том же порядке, в котором перечислены соответствующие им пары во входном файле.

Пример

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

Задача В. 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 С. Учиться! - HARD

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.8 секунд
Ограничение по памяти:	256 мегабайт

Каждый год огромное количество выпускников, сдавшие ЕГЭ, выбирают, куда же они пойдут учиться. Не удивительно, что многие из них предпочитают перебраться поближе к столице. Транспортная инфраструктура страны переживает не лучшие времена, и в приемлемом качестве поддерживается минимально возможное число городов, необходимое для того, чтобы от любого города можно было добраться до любого другого.

Каждый выпускник оценивает свои результаты сдачи экзаменов, и решает, насколько далеко от своего родного города в сторону столицы он сможет уехать.

Выпускников настолько много, что вам не требуется выводить для каждого из них, до какого города он сможет доехать. Достаточно вывести сумму ответов для каждого выпускника.

Запросы генерируются следующим образом. Заданы числа a_1, a_2 и числа x, y и z . Числа a_3, \dots, a_{2m} генерируются следующим образом: $a_i = (x \cdot a_{i-2} + y \cdot a_{i-1} + z) \bmod n$. Первый запрос имеет вид $\langle a_1, a_2 \rangle$. Если ответ на $i - 1$ -й запрос равен v , то i -й запрос имеет вид $\langle (a_{2i-1} + v) \bmod n, a_{2i} \rangle$. В i -м запросе первое число соответствует городу, в котором окончил школу i -й выпускник, а второе — насколько далеко от родного города он может уехать. Все выпускники стараются перебраться как можно ближе к столице.

Формат входных данных

Первая строка содержит два числа: n ($1 \leq n \leq 100\,000$) и m ($1 \leq m \leq 10\,000\,000$). Столица имеет номер 0. Вторая строка содержит $n - 1$ целых чисел, i -е из этих чисел равно номеру следующего за городом i на пути к столице. Третья строка содержит два целых числа в диапазоне от 0 до $n - 1$: a_1 и a_2 . Четвертая строка содержит три целых числа: x, y и z , эти числа неотрицательны и не превосходят 10^9 .

Формат выходных данных

Выведите в выходной файл сумму номеров городов — ответов на все запросы.

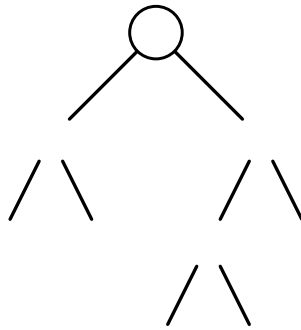
Примеры

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

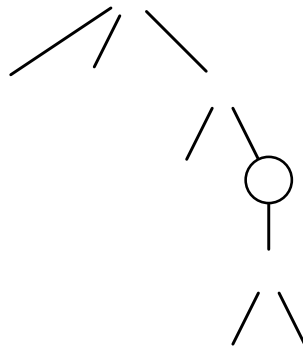
Задача D. Dynamic LCA

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Постановка задачи о *наименьшем общем предке* такова: дано дерево T с выделенным корнем и две вершины u и v , $\text{lca}(u, v)$ — вершина с максимальной глубиной, которая является предком u , и v . Например, на картинке внизу $\text{lca}(8, 7)$ — вершина 3.



С помощью операции $\text{chroot}(u)$ мы можем менять корень дерева, достаточно отметить u , как новый корень, и направить ребра вдоль пути от корня. Наименьшие общие предки вершин поменяются соответственно. Например, если мы сделаем $\text{chroot}(6)$ на картинке сверху, $\text{lca}(8, 7)$ станет вершина 6. Получившееся дерево изображено внизу.



Вам дано дерево T . Изначально корень этого дерева — вершина 1. Напишите программу, которая поддерживает эти две операции: $\text{lca}(u, v)$ и $\text{chroot}(u)$.

Формат входных данных

Входной файл состоит из нескольких тестов.

Первая строка каждого теста содержит натуральное число n — количество вершин в дереве ($1 \leq n \leq 100\,000$). Следующие $n - 1$ строк содержат по 2 натуральных числа и описывают ребра дерева. Далее идет строка с единственным натуральным числом m — число операций. Следующие m строк содержат операции. Строка $? u v$ означает операцию $\text{lca}(u, v)$, а строка $! u$ — $\text{chroot}(u)$. Последняя строка содержит число 0.

Сумма n для всех тестов не превосходит 100 000. Сумма m для всех тестов не превосходит 200 000.

Формат выходных данных

Для каждой операции $? u v$ выведите значение $\text{lca}(u, v)$. Числа разделяйте переводами строк.

Пример

стандартный ввод	стандартный вывод
9	2
1 2	1
1 3	3
2 4	6
2 5	2
3 6	3
3 7	6
6 8	2
6 9	
10	
? 4 5	
? 5 6	
? 8 7	
! 6	
? 8 7	
? 4 5	
? 4 7	
? 5 9	
! 2	
? 4 3	
0	

Задача Е. 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:	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 G. 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 Н. 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 листьев 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

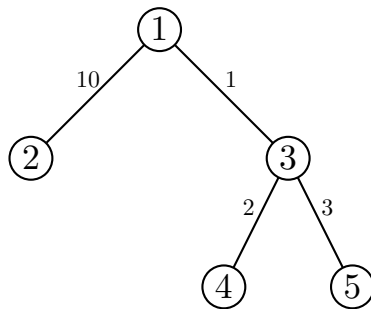
Выведите q чисел — ответы на запросы в порядке, в котором они заданы во входных данных.

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. XOR на дереве

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

Дано взвешенное дерево из n вершин. Гарантируется, что n нечётное. Можно сколько угодно раз сделать следующую операцию:

- Выбрать ребро (v, u) , предположим, что его текущий вес равен w . Тогда надо изменить вес всех рёбер, инцидентных ровно одной из вершин v и u , на их вес XOR w . То есть для каждого ребра (a, b) веса w_1 , что ровно одна из вершин a и b равна v или u , надо заменить w_1 на w_1 XOR w .

Для каждого ребра известен его необходимый финальный вес. Определите можно ли такими операциями этого добиться.

Формат входных данных

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

Следующие $n - 1$ строк содержат описания рёбер. i -я строка содержит 4 числа v_i, u_i, w_i и f_i ($1 \leq v_i, u_i \leq n, v_i \neq u_i, 1 \leq w_i, f_i < 2^{30}$) — концы ребра, а так же начальный и финальный вес соответственно.

Формат выходных данных

Выведите YES, если можно добиться таких весов, и NO иначе.

Примеры

стандартный ввод	стандартный вывод
3 1 2 1 1 2 3 8 9	YES
5 1 2 0 3 1 3 1 0 1 4 2 1 1 5 0 0	NO
1	YES

Задача J. 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 К. 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 L. Дерево

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

Дано дерево из n вершин и q запросов.

Каждый запрос начинается с трех целых чисел k , m и r , и продолжается k вершинами дерева a_1, a_2, \dots, a_k . Чтобы ответить на запрос, предположите, что дерево подвешено за вершину r . Рассмотрим разбиения данных k вершин на **не более чем** m групп так, что выполняются следующие условия:

- Каждая вершина принадлежит ровно одной группе, каждая группа содержит хотя бы одну вершину.
- Ни в одной группе нет двух вершин таких, что одна является предком (не обязательно непосредственным) другой.

Выведите количество различных таких разбиений по модулю $10^9 + 7$ для каждого запроса.

Формат входных данных

Первая строка содержит два целых числа n и q ($1 \leq n, q \leq 10^5$) — количество вершин в дереве и количество запросов, соответственно.

Каждая из следующих $n - 1$ вершин содержит два целых числа u и v ($1 \leq u, v \leq n, u \neq v$), обозначающие ребро между вершинами u и v . Гарантируется, что данный граф является деревом.

Каждая из следующих q строк начинается с трех целых чисел k , m и r ($1 \leq k, r \leq n, 1 \leq m \leq \min(300, k)$) — количество вершин, максимальный размер группы и корень дерева для данного запроса, соответственно. После этого следуют k различных целых чисел a_1, a_2, \dots, a_k ($1 \leq a_i \leq n$) — вершины текущего запроса.

Гарантируется, что сумма значений k по всем запросам не превосходит 10^5 .

Формат выходных данных

Выведите q строк, где i -я строка содержит ответ на i -й запрос.

Примеры

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

Замечание

Рассмотрим первый пример.

В первом запросе нужно разделить три данные вершины (7, 4 и 3) на не более чем три группы, считая, что корнем дерева является вершина 2. Когда дерево подвешено за вершину 2, вершина 4 является предком вершин 3 и 7. Поэтому нельзя все вершины отнести к одной группе. Есть только 1 способ разделить эти вершины на две группы: [4] и [3, 7]. Кроме того, есть один способ разделить данные вершины на три группы: [7], [4] и [3]. Таким образом, есть всего 2 способа разбить данные вершины на не более чем три группы.

Во втором запросе дерево подвешено за вершину 4, при этом 6 является предком 2, а 2 является предком 1. Поэтому нельзя все вершины отнести к одной группе.

Задача М. Древландия

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	512 мегабайт

В древландии есть города, и первый город — *столица*. Города соединены автобусными маршрутами так, что для города $i \neq 1$ есть автобус, который идёт оттуда в город p_i ($p_i < i$), именно в таком направлении.

В стране есть национальные блюда. Каждый город имеет своё специальное блюдо, оно и только оно может быть куплено там. Тип специального блюда для каждого города — один из типов национальных блюд.

Несколько друзей из нескольких городов хотят встретиться в одном городе для вечеринки. Они выбирают город такой, что, если они одновременно начнут туда идти, они встретятся там быстро, как только возможно. Путешествие на автобусе требует 1 единицу времени.

Они хотят купить некоторые блюда для вечеринки, соблюдая следующие требования:

1. Каждый друг должен купить одно и то же количество блюд.
2. Не должно быть двух блюд одного типа на вечеринке.
3. Каждый друг может купить только блюда, соответствующие городам, которые он посетил.

Для заданных запросов, найдите максимальное количество блюд, которое может быть на вечеринке.

Формат входных данных

Первая строка содержит три числа n , m , q .

- n — количество городов
- m — количество типов национальных групп
- q — количество запросов

Вторая строка содержит $n - 1$ чисел p_2, \dots, p_n , описывающие автобусные маршруты.

Третья строка содержит n чисел a_1, \dots, a_n , описывающие типы блюд, продающиеся в соответствующих городах.

Следующие строки содержат описания запросов. Каждый запрос описывается в следующем формате: число c , обозначающее количество друзей, а затем c чисел v_1, \dots, v_c . Пусть ответ на предыдущий ответ равен X (для первого запроса $X = 0$). Тогда друзья находятся в вершинах $(v_1 - 1 + X) \bmod n + 1, \dots, (v_c - 1 + X) \bmod n + 1$.

Не гарантируется, что для конкретного запроса все v_i различны.

- $2 \leq n \leq 3 \cdot 10^5$
- $1 \leq m \leq 1000$
- $1 \leq q \leq 5 \cdot 10^4$
- $1 \leq p_i < i$
- $1 \leq a_i \leq m$
- $2 \leq c \leq 5$
- $1 \leq v_i \leq n$

Формат выходных данных

Выведите q строк, i -я из которых — ответ на i -й запрос.

Примеры

стандартный ввод	стандартный вывод
5 3 4 1 2 2 1 2 3 1 3 1 2 3 4 3 5 2 2 4 3 4 2 5 2 2 2	2 0 0 0
11 6 3 1 2 2 4 5 4 5 8 9 4 5 6 1 1 2 3 2 3 4 5 2 3 3 10 8 4 6 5 10 10 2 9 6	6 4 2

Задача N. 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.

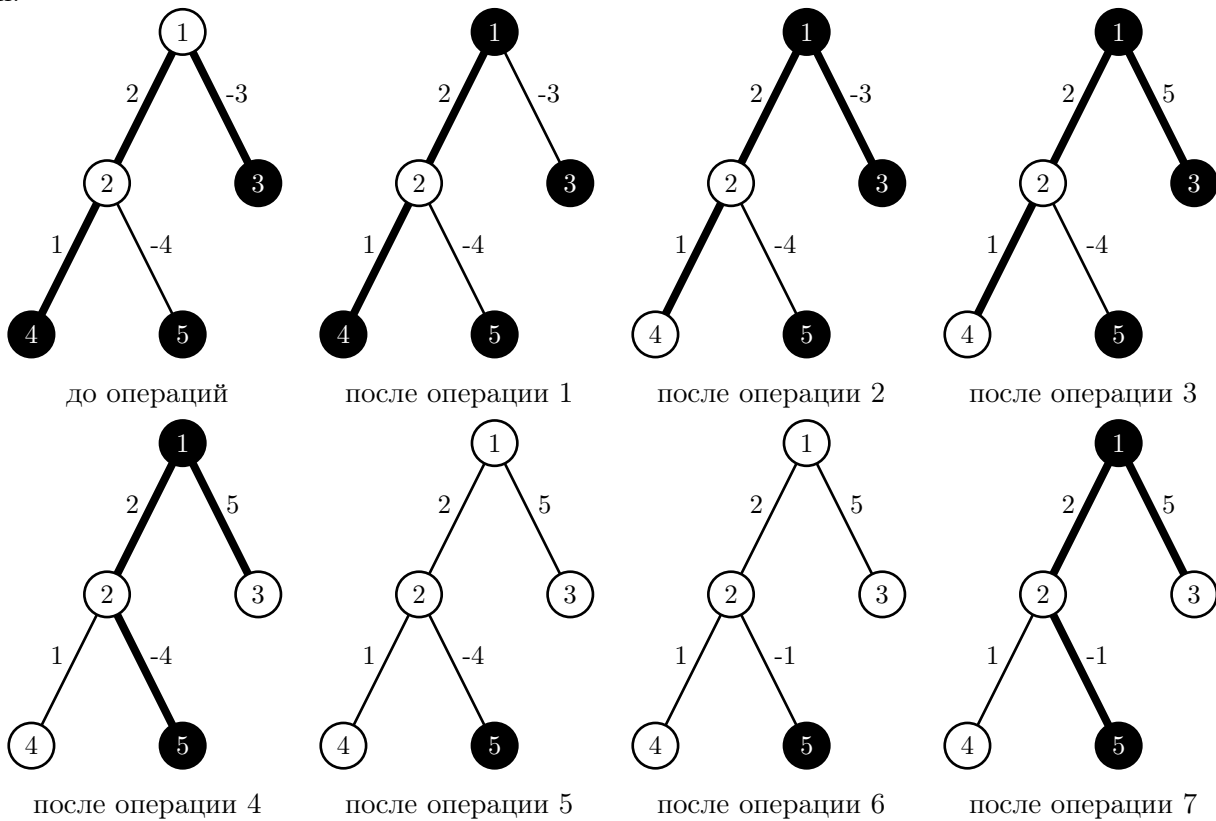
Подзадача	Баллы	Ограничения		
		n	q	Additional
1	3	$n \leq 20$	$q \leq 20$	—
2	3	$n \leq 500$	$q \leq 500$	—
3	3	$n \leq 100$	$q \leq 5000$	—
4	5	$n \leq 5000$	$q \leq 5000$	—
5	7	$n \leq 10^5$	$q \leq 10^5$	$D \leq 30, H \leq 20$
6	20	$n \leq 10^5$	$q \leq 10^5$	$H \leq 20$
7	10	$n \leq 10^5$	$q \leq 10^5$	$p_v = v - 1$ 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.



Задача О. Соединение и разъединение

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Вы когда-нибудь слышали про обход в глубину? Например, используя этот алгоритм, вы можете проверить является ли граф связным за время $O(E)$. Вы можете даже посчитать количество компонент связности за то же время.

А вы когда-нибудь слышали про систему непересекающихся множеств? Используя эту структуру, вы можете быстро обрабатывать запросы “Добавить ребро в граф” и “Посчитать количество компонент связности в графе”.

А вы когда-нибудь слышали о *динамической* задаче связности? В этой задаче вам необходимо обрабатывать три типа запросов:

1. Добавить ребро в граф.
2. Удалить ребро из графа.
3. Посчитать количество компонент связности в графе.

Можно считать, что граф является неориентированным. Изначально граф является пустым.

Формат входных данных

В первой строке находятся два целых числа N и K — количество вершин и количество запросов, соответственно ($1 \leq N \leq 300\,000$, $0 \leq K \leq 300\,000$). Следующие K строк содержат запросы, по одному в строке. Каждый запрос имеет один из трех типов:

1. $+ u v$: Добавить ребро между вершинами u и v . Гарантируется, что такого ребра нет.
2. $- u v$: Удалить ребро между u и v . Гарантируется, что такое ребро есть.
3. $?$: Посчитать количество компонент связности в графе.

Вершины пронумерованы целыми числами от 1 до N . Во всех запросах $u \neq v$.

Формат выходных данных

Для каждого запроса типа ‘?’, Выведите количество компонент связности в момент запроса.

Пример

стандартный ввод	стандартный вывод
5 11	5
?	1
+ 1 2	1
+ 2 3	2
+ 3 4	
+ 4 5	
+ 5 1	
?	
- 2 3	
?	
- 4 5	
?	

Замечание

Решения, не работающие в online, будут получать reject.

Задача Р. Интерактивная вершина

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Это интерактивная задача

У Ильдара есть дерево из n вершин и он показал его вам. Он выбирает одну вершину u как специальную вершину, но он не сообщает вам ничего о ней!

Вместо этого, вы можете задавать ему вопросы. Для каждого вопроса вы должны выбрать вершину x , натуральное число k и k вершин v_1, v_2, \dots, v_k и он вам скажет, правда ли, что $\min(\text{dist}(u, v_i)) \geq \text{dist}(u, x)$. Здесь, $\text{dist}(p, q)$ это количество ребер на простом пути между вершинами p и q в дереве.

Вы должны угадать специальную вершину за не больше, чем $4 \lceil \log_2 n \rceil$ вопросов.

Ильдар очень добрый, поэтому он не будет менять специальную вершину между вашими вопросами (другими словами, интерактор неадаптивный).

Поскольку ограничения большие и flush это тяжелая операция, убедитесь, что вы не делаете операцию flush слишком часто. Рекомендуется делать flush только после вывода каждого вопроса.

Протокол взаимодействия

Процесс взаимодействия начинается с того, что на первой строке вводится целое число n : количество вершин в дереве Ильдара ($2 \leq n \leq 200\,000$).

Каждая из следующих $n - 1$ строк содержит два целых числа u и v ($1 \leq u, v \leq n$), означающих ребро между u и v . Гарантируется, что данные ребра образуют дерево.

После этого, вы можете задавать вопросы.

Чтобы задать вопрос, выведите одну строку, содержащую “? k ” ($1 \leq k \leq n$), целое число x ($1 \leq x \leq n$) и затем k различных целых чисел v_1, v_2, \dots, v_k ($1 \leq v_i \leq n$). Разделяйте соседние числа в строке ровно одним пробелом. Затем сделайте flush выходного потока.

После каждого вопроса, считайте одно целое число $ans \in \{0, 1\}$. Если $\min(\text{dist}(u, v_i)) \geq \text{dist}(u, x)$, тогда ans будет равно 1. Иначе, ans будет равно 0.

Когда вы нашли специальную вершину u ($1 \leq u \leq n$), выведите одно целое число “! u ”, сделайте flush выходного потока и завершите работу программы.

Ваше решение получит Wrong Answer или Time Limit Exceeded если вы сделаете больше чем $4 \lceil \log_2 n \rceil$ вопросов.

Ваше решение получит Idleness Limit Exceeded если оно не совершает никаких действий или не делает flush выходного потока.

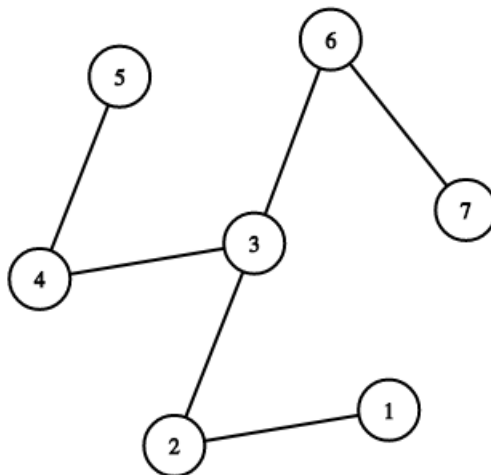
Чтобы сделать flush выходного потока, вы можете использовать:

- `fflush(stdout)` или `cout.flush()` в C++;
- `System.out.flush()` в Java;
- `flush(output)` в Pascal;
- `stdout.flush()` в Python;
- используйте документацию других языков.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 1 2 1 3 1 4 1 5 1	? 4 1 2 3 4 5 ! 1
5 1 2 1 3 1 4 1 5 0 0 0 0	? 4 1 2 3 4 5 ? 3 1 2 3 4 ? 2 1 2 3 ? 1 1 2 ! 2
7 1 2 2 3 3 4 4 5 3 6 6 7 1	? 3 3 5 7 1 ! 3

Замечание



Задача Q. 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.

Tinkoff Generation 2022-2023. А. Структуры на деревьях
Водный стадион, 26 ноября 2022

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	Offline-проверка.
10	3	$n \leq 500\,000$	0 – 9	Offline-проверка.
11	2	$n \leq 750\,000$	0 – 10	Offline-проверка.
12	1	–	0 – 11	Offline-проверка.