

Задача А. 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

Лупа и Пупа пошли получать зарплату. Но в бухгалтерии опять всё перепутали. Лупа получил зарплату за Пупу, а Пупа ...

Пупа не хочет получать за Лупу и хочет доказать бухгалтерии, что она не права.

Пупа 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 Пупе. Пупа 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 Пупе.

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. Пупа 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 Пупы в каждый из дней с 0-го по m -й. Напоминаем, что Пупа 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 1 2 1 1 0 1 1 3 2 3	1 6 31 100

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 В. Декомпозиция

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

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

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

Формат входных данных

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

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

Формат выходных данных

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

Примеры

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

Задача С. 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 D. 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 1 2 1 4 1 1 1 10 1 1 0 20 2 1 2 2	20 10
5 1 2 30 1 3 50 3 4 70 3 5 60 8 1 3 72 6 2 5 1 4 60 5 2 3 2 2 1 2 144 7 2 4 2 5	6 6 0 5 7

Задача Е. Ландыши.

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

Ты и так, ведь, хороша! Развернись, моя душа, ну зачем нам эти ландыши?...

Вашей девушке очень нравятся ландыши, поэтому вместо того, чтобы спокойно поспать, вы обдумываете, в какой из дней лучше идти собирать ландыши. Ландыши можно купить в n цветочных магазинах, в i -м магазине изначально находится a_i ландышей. Если в какой-то из дней в i -м магазине остаются ландыши, то этот магазин передает все свои ландыши в магазин p_i ($1 \leq p_i < i$), магазин 1 все оставшиеся ландыши выбрасывает на помойку.

У вас есть q возможных планов, каждый из которых выглядит следующим образом:

- В день t вы посещаете k магазинов $v, p[v], p[p[v]], \dots$ и скупаете там все ландыши.

Никому кроме вас эти ландыши не нужны, и план надежен как швейцарские часы, но есть одна проблема. На этапе планирования вы не справились посчитать, сколько ландышей получится купить в каждом из планов, поэтому вам надо написать программу, которая решает поставленную задачу.

Формат входных данных

В первой строке указаны числа n и q ($2 \leq n \leq 200\,000$, $1 \leq q \leq 200\,000$).

Во второй строке указаны числа $p_2, \dots, p_n$ ($1 \leq p_i < i$).

Далее следуют q планов, каждый из которых описывается тройкой чисел t, v, k ($1 \leq t, v, k \leq n$). Гарантируется, что k не превосходит количества магазинов, в которые по цепочке могут попасть ландыши из магазина v .

Формат выходных данных

Выведите q чисел — количество ландышей, которое вы можете приобрести для каждого из планов.

Пример

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

Задача F. Keep Them Equal

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

Перед вами дерево — связный граф с n вершинами и $n - 1$ ребром.

Когда из дерева удаляется вершина, то оно распадается на несколько деревьев.

Для каждой вершины вы должны найти вторую вершину такую, что если эти две вершины удалить из дерева, то размер наибольшего из получившихся деревьев будет минимально возможным. Размером дерева считается количество вершин в этом дереве.

Формат входных данных

Первая строка содержит целое число n — количество вершин ($2 \leq n \leq 300\,000$).

Следующие $n - 1$ строки содержат по паре целых чисел a_i и b_i , означающих, что данные вершины соединены ребром ($1 \leq a_i, b_i \leq n$).

Формат выходных данных

Для каждого $1 \leq i \leq n$ выведите в отдельной строке номер требуемой вершины (то есть вершины, которая должна быть удалена вместе с i -й).

Разрешается выводить любую подходящую вершину.

Пример

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

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. Ядовитый лабиринт

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

ВThego должен выбраться из лабиринта, который представляет собой дерево с n вершинами и $n - 1$ ребрами, где каждое ребро имеет свою длину. Также в лабиринте находится m различных видов ядов, каждый из которых находится в двух вершинах.

Если ВThego попадает на вершину, он тут же отравляется всеми ядами в этой вершине. Если он оказывается на вершине, где уже был ранее, он не будет отравлен повторно.

Чтобы вылечиться от яда, ВThego должен снова быть отравлен тем же видом яда. Для этого он должен посетить вершину с таким же типом яда.

ВThego начинает свой путь с вершины s , где он тут же отравляется всеми ядами в этой вершине. Затем он проходит некоторые вершины, пока не избавится от ядов, после чего возвращается в вершину s и покидает лабиринт.

Необходимо найти стартовую вершину s , такую что если ВThego начнет свой путь с этой вершины, то он пройдет минимальное расстояние, при условии, что он выберет оптимальный маршрут

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 2 \cdot 10^5$) - количество вершин в лабиринте и количество видов ядов.

Следующие $n - 1$ строк содержат по три целых числа u_i , v_i и w_i ($1 \leq u_i, v_i \leq n, 1 \leq w_i \leq 10^9$) - ребро между вершинами u_i и v_i с весом w_i .

Следующие m строк содержат по два целых числа a_i и b_i ($1 \leq a_i, b_i \leq n$) - вершины, где находится яд i .

Формат выходных данных

Выведите одно целое число - минимальное расстояние, которое ВThego должен пройти, чтобы излечиться от всех ядов, начиная из оптимальной стартовой вершины.

Примеры

стандартный ввод	стандартный вывод
4 2 1 2 1 1 3 10 1 4 100 1 3 2 4	20
5 2 1 2 1 1 3 10 1 4 100 1 5 1000 1 3 2 4	0
7 4 1 2 8 1 3 9 2 4 10 2 5 11 3 6 12 3 7 13 2 3 7 6 2 1 4 5	34

Задача I. Окружения путей

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

Дано дерево на N вершинах, пронумерованных последовательными целыми числами от 1 до N .

Простым путём P между вершинами a и b называется такая последовательность из k вершин ($a = P_1, P_2, \dots, P_k = b$), что любые две соседние вершины соединены ребром и каждая вершина присутствует в пути не более одного раза. Заметим, что a и b могут быть равны. Считается, что ребро лежит на простом пути, если оно соединяет две соседние вершины этого пути.

Окружение простого пути P состоит из всех рёбер, у которых ровно одна вершина принадлежит P .

Каждое ребро в дереве может быть заблокировано или разблокировано. Изначально все рёбра заблокированы. Напишите программу, которая поддерживает запросы двух типов:

- 1. Считает количество заблокированных рёбер на простом пути между вершинами a и b .
- 2. Разблокирует все рёбра, лежащие на простом пути между a и b и блокирует все рёбра из окружения данного пути.

Формат входных данных

Первая строка входных данных содержит одно целое число N — количество вершин N ($1 \leq N \leq 200\,000$).

Каждая из последующих $N - 1$ строк содержит описание ребра в формате $a_i b_i$. Гарантируется, что заданный граф является деревом.

Следующая строка содержит количество запросов Q ($1 \leq Q \leq 300\,000$).

Оставшиеся Q строк содержат запросы в формате $t_i a_i b_i$ — тип запроса и конечные точки, соответственно.

Формат выходных данных

Для каждого запроса типа 0 выведите ответ на запрос в отдельной строке.

Пример

стандартный ввод	стандартный вывод
19	2
1 2	3
2 3	2
1 5	2
5 4	
5 6	
6 7	
6 8	
1 11	
11 12	
11 13	
11 10	
10 9	
13 14	
13 15	
15 16	
15 17	
15 18	
15 19	
6	
1 19 8	
0 16 2	
0 16 3	
1 12 9	
0 19 8	
0 16 9	

Задача J. Префиксные максимумы

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

Дано взвешенное дерево на n вершинах.

Обозначим за $w(i, j)$ вес ребра между вершинами i и j .

Рассмотрим простой путь $P = (u, s_1, \dots, s_{t-1}, v)$ от вершины u до вершины v . Обозначим последовательность весов рёбер на пути P , как $a = (a_1, a_2, \dots, a_t)$, где $a_1 = w(u, s_1)$, $a_2 = w(s_1, s_2)$, $\dots$, $a_t = w(s_{t-1}, v)$.

Пусть $f(u, v) = \sum_{i=1}^t \max_{j=1..i} \{a_j\}$, то есть сумма префиксных максимумов a .

Даны q запросов, заданные номерами вершин u и v , посчитайте для них $f(u, v)$.

Формат входных данных

В первой строке заданы два числа n и q ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq q \leq 10^6$) — кол-во вершин в дереве и кол-во запросов, соответственно.

Каждая из следующих $n - 1$ строк содержит три числа a_i , b_i и $w(a_i, b_i)$ ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$, $1 \leq w(a_i, b_i) \leq 10^9$) — описание рёбер дерева.

Каждая из следующих q строк содержит два числа u_i и v_i ($1 \leq u_i, v_i \leq n$) — параметры i -го запроса.

Пример

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

Задача К. Побег кроликов

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	1024 мегабайта

Кроличья нора состоит из n комнат, пронумерованных целыми числами от 1 до n . Комната номер 1 напрямую соединена с выходом из норы. Также задано $n - 1$ пара комнат (u, v) , таких что комната u соединена проходом с комнатой v . Гарантируется, что между любой парой комнат существует путь по этим проходам.

В i -й норе живёт кролик с номером i и прыгучестью i , что позволяет ему за один прыжок пропрыгать не более i проходов. Кролики начинают выбегать из норы в порядке их номеров. При побеге кролики могут прыгать из комнаты только в комнату, которая **строго ближе** к выходу. При побеге кролика с прыгучестью i , если не существует i комнат строго ближе к выходу, то он может сразу выпрыгнуть из норы.

В некоторых комнатах будет проводится уборка, и так как прыгать на мокрый пол опасно, прыгать в эти комнаты кролики не будут. Заметим, что если кролик уже находится в этой комнате, прыгать из неё ему ничего не мешает.

Перед началом побега i -го кролика, состояние не более одной из комнат может поменяться.

Для каждого кролика, найдите минимальное число прыжков, которое ему надо сделать, чтобы выпрыгнуть из норы, или определите, что это невозможно.

Формат входных данных

В первой строке задано число n ($1 \leq 5 \cdot 10^5$) — количество кроликов.

В следующих $n - 1$ строках задано по два числа u, v ($1 \leq u, v \leq n$) — описание проходов.

В следующей строке задано n чисел $b_1, \dots, b_n$ ($b_i \in \{0, 1\}$), где если $b_i = 0$, значит в комнате ведётся уборка и в неё нельзя прыгать, и $b_i = 1$ иначе.

В следующей строке задано n чисел $c_1, \dots, c_n$ ($0 \leq c_i \leq n$), где c_i обозначает номер комнаты состояние которой меняется перед началом побега i -го кролика. Если $c_i = 0$, то никакая комната не меняет состояние.

Формат выходных данных

Выведите n чисел, где i -е равно минимальному кол-ву прыжков, которое нужно i -му кролику, чтобы выпрыгнуть из норы, или -1 , если он не может выпрыгнуть.

Примеры

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