

## 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

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 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	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 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  $n - 1$  строк 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. Гигантский Пингвин

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



Pengsoo — крайне популярный гигантский Корейский пингвин. Он очень невоспитанный и любит пить.

Сейчас Pengsoo очень занят — он отвечает на запросы в очередной задаче на графы.

У него есть связный неориентированный граф, в котором каждая **вершина** лежит не более, чем на  $k$  вершинно-простых циклах.

Он хочет отвечать на запросы двух типов.

- Пометить вершину  $v$ .
- Найти ближайшую помеченную вершину для вершины  $v$  (гарантируется, что на момент данного запроса в графе имеется хотя бы одна помеченная вершина).

Pengsoo очень ленивый, поэтому он решил вздремнуть и попросил вас ответить на данные запросы. Если вы не справитесь до того, как он проснется, он будет над вами издеваться, так что поторопитесь!

### Формат входных данных

В первой строке записаны три числа  $n, m, k$  ( $1 \leq n \leq 100\,000$ ,  $n - 1 \leq m \leq 200\,000$ ,  $0 \leq k \leq 10$ ) — количество вершин, ребер и максимальное количество вершинно-простых циклов, проходящих через одну вершину.

В следующих  $m$  строках содержится описание ребер. В  $i$ -й строке записаны два числа  $u_i, v_i$  ( $1 \leq u_i, v_i \leq n$ ,  $u_i \neq v_i$ ), означающие, что в графе есть ребро между вершинами  $u_i$  и  $v_i$ .

Гарантируется, что в графе нет красных ребер, граф является связным и каждая вершина лежит не более, чем на  $k$  вершинно-простых циклах.

В следующей строке записано число  $q$  ( $1 \leq q \leq 200\,000$ ) — количество запросов.

В каждой из следующих  $q$  строк содержится описание запроса. В  $i$ -й строке записаны два числа  $t_i, v_i$  ( $1 \leq t_i \leq 2$ ,  $1 \leq v_i \leq n$ ).

Если  $t_i = 1$ , пометьте вершину  $v_i$ . Гарантируется, что данная вершина не была помечена ранее.

Если  $t_i = 2$ , найдите расстояние до ближайшей помеченной вершины от вершины  $v_i$ . Гарантируется, что в графе уже есть хотя бы одна помеченная вершина.

### Формат выходных данных

Для каждого запроса с  $t_i = 2$  выведите расстояние до ближайшей помеченной вершины.



## Примеры

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

## Задача Н. 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 I. Ядовитый лабиринт

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

ВТhero 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 ВТhero 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, ВТhero должен снова быть отравлен тем же видом яда. Для этого он должен посетить вершину с таким же типом яда.

ВТhero 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 ВТhero 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 ВТhero 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