

## Задача А. Наименьший общий предок

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

У Бобо есть корневое дерево из  $n$  вершин, удобно пронумерованных числами  $1, 2, \dots, n$ . Вершина  $1$  — корень дерева, и  $i$ -я вершина имеет вес  $w_i$ .

Он хотел бы посчитать  $f(2), f(3), \dots, f(n)$  где

$$f(i) = \sum_{j=1}^{i-1} w_{\text{LCA}(i,j)}.$$

### Формат входных данных

Входные данные содержит ноль или более тестовых примеров и заканчиваются символом конца файла. Для каждого тестового примера:

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

Вторая строка содержит  $n$  чисел  $w_1, w_2, \dots, w_n$  ( $1 \leq w_i \leq 10^4$ ).

Третья строка содержит  $(n - 1)$  чисел  $p_2, p_3, \dots, p_n$ , где  $p_i$  обозначает ребро из вершины  $p_i$  в вершину  $i$  ( $1 \leq p_i \leq n$ ). Ребра образуют дерево.

Гарантируется, что сумма всех  $n$  не превосходит  $2 \cdot 10^5$ .

### Формат выходных данных

Для каждого тестового примера, выведите  $n - 1$  чисел:  $f(2), f(3), \dots, f(n)$ .

### Примеры

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

## Задача В. Адам и дерево

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

Когда у Адама появляется корневое дерево (связный неориентированный граф без циклов), он сразу начинает его раскрашивать. Более формально, каждому ребру он сопоставляет некоторый цвет таким образом, чтобы выполнялись два условия:

- Не существует вершины, у которой больше двух инцидентных ребер покрашены в один цвет.
- Для любых двух вершин, у которых есть инцидентные ребра, покрашенные в один цвет (скажем,  $c$ ), путь между ними содержит ребра только цвета  $c$ .

Не все раскраски дерева нравятся Адаму одинаково. Рассмотрим путь от некоторой вершины до корня. Количество различных цветов на этом пути назовем стоимостью вершины. Стоимостью раскраски дерева будем называть максимальную стоимость среди всех вершин дерева. Помогите Адаму определить минимальную стоимость раскраски дерева.

Изначально дерево Адама состоит из одной вершины, которая имеет номер один и является корнем. За один ход Адам подвешивает к уже существующей вершине новую, которая получает номер, равный наименьшему положительному целому не занятому числу. После каждой операции вам нужно сообщать минимальную стоимость раскраски получившегося дерева.

### Формат входных данных

В первой строке задано целое число  $n$  ( $1 \leq n \leq 10^6$ ) — количество добавлений новых вершин. Во второй строке записано  $n$  чисел  $p_i$  ( $1 \leq p_i \leq i$ ) — номера вершин, к которым подвешивают очередную вершину.

### Формат выходных данных

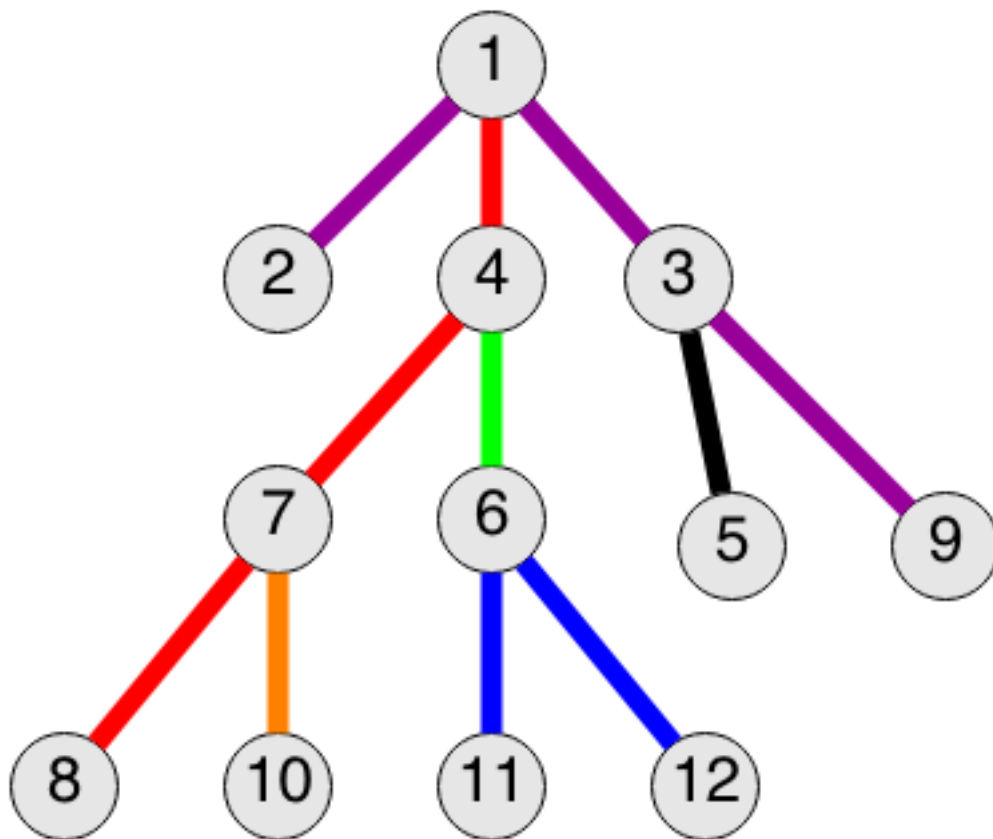
Выведите  $n$  целых чисел — минимальные стоимости раскраски дерева после каждого добавления.

### Примеры

стандартный ввод	стандартный вывод
11 1 1 1 3 4 4 7 3 7 6 6	1 1 1 1 1 2 2 2 2 3

### Замечание

На картинке изображен один из возможных вариантов раскраски дерева из примера в самый последний момент. Стоимость вершин с номерами 11 и 12 равна трем.



## Задача С. Почтовая реформа

Имя входного файла:	mail.in
Имя выходного файла:	mail.out
Ограничение по времени:	4 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

mail.in	mail.out
3 1 2 3 1 3 2 3 5 ? 1 2 ! 1 5 ? 2 3 ! 3 2 ? 1 2	3 3 5
1 100 5 ! 1 1 ? 1 1 ! 1 1000 ? 1 1 ! 1 1	1 1000

## Задача D. $k$ -е ребро в поддереве

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

Рассмотрим дерево из  $n$  вершин, пронумерованных числами от 1 до  $n$ . На каждом ребре написано число от 1 до  $10^9$ .

Вам нужно ответить на следующие запросы: для некоторого поддерева, выпишите значения, написанные на ребрах и найдите  $k$ -е среди них. Каждое поддерево задается подмножеством из  $z$  вершин. Возьмем объединение кратчайших путей между всеми парами этих вершин. Иначе говоря, ребро принадлежит поддереву тогда и только тогда, когда оно лежит на хотя бы одном кратчайшем пути между парой заданных вершин.

### Формат входных данных

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

Следующие  $n - 1$  строк содержат по три числа каждая;  $i$ -я из них содержит числа  $a_i$ ,  $b_i$  и  $c_i$ : номера вершины, соединенных  $i$ -м ребром, и число, которое написано на этом ребре ( $1 \leq a_i, b_i \leq n$ ,  $a_i \neq b_i$ ,  $1 \leq c_i \leq 10^9$ ).

Следующая строка содержит число  $q$ : количество запросов ( $1 \leq q \leq 10^5$ ).

Каждая из следующих  $q$  строк описывает очередной запрос. Множество из  $z$  вершин задается  $z$  различными числами  $v_1, v_2, \dots, v_z$ , после которых записан 0 ( $1 \leq z \leq n$ ,  $1 \leq v_j \leq n$ ). После этого записано число  $k$  ( $1 \leq k \leq n - 1$ ). Смотрите примеры в условии для пояснения.

Сумма  $z$  по всем запросам не превосходит  $10^5$ .

### Формат выходных данных

Для каждого запроса выведите ответ в отдельной строке: значение  $k$ -го ребра в заданном поддереве. Если в поддереве меньше, чем  $k$  ребер, выведите  $-1$ .

### Примеры

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

## Задача Е. БДБД

Имя входного файла:	lwdb.in
Имя выходного файла:	lwdb.out
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	256 мегабайт

Большая Древесная База Данных создана для того, чтобы в ней можно было надежно сохранить и раскрасить любое дерево. В новой версии БДБД запланирован новый функционал, для реализации которого потребуется вновь переосмыслить теорию графов.

В БДБД хранится взвешенное дерево. В языке запросов Системы Управления Большой Древесной Базы Данных (СУБДБД) предусмотрены два вида запросов:

- «1  $v$   $d$   $c$ » — покрасить все вершины, находящиеся на расстоянии не более  $d$  от вершины  $v$ , в цвет  $c$ . Все вершины изначально окрашены в цвет с номером 0.
- «2  $v$ » — вывести цвет вершины  $v$ .

Необходимо запрограммировать работу СУБДБД и ответить на все запросы пользователя.

### Формат входных данных

В первой строке число  $N$  ( $1 \leq N \leq 10^5$ ) — количество вершин дерева.

Следующие  $N - 1$  строк содержат описание ребер, по три числа в строке  $a_i, b_i, w_i$  ( $1 \leq a_i, b_i \leq N$ ,  $a_i \neq b_i$ ,  $1 \leq w_i \leq 10^4$ ), где  $i$ -ое ребро имеет вес  $w_i$  и соединяет вершины  $a_i$  и  $b_i$ .

В следующей строке число  $Q$  ( $1 \leq Q \leq 10^5$ ) — число запросов. В каждой из  $Q$  следующих строк запросы одного из двух видов:

- Числа 1,  $v$ ,  $d$ ,  $c$  ( $1 \leq v \leq N$ ,  $0 \leq d \leq 10^9$ ,  $0 \leq c \leq 10^9$ ).
- Числа 2,  $v$  ( $1 \leq v \leq N$ ).

Все числа во входных данных целые.

### Формат выходных данных

Для каждого запроса второго типа необходимо вывести в отдельной строке цвет запрошенной вершины.

### Примеры

lwdb.in	lwdb.out
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 F. В бухгалтерии опять всё перепутали

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 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 G. Уголки

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

После почти повсеместного закрытия казино у нас в стране, предприимчивые дельцы стали создавать клубы по игре в различные азартные, но пока еще не запрещенные игры. Одной из таких игр стала игра «Уголки».

Игроку выдается карточка, на которой нарисована таблица, состоящая из  $M$  строк и  $N$  столбцов, каждым элементом которой является целое число. Строки и столбцы пронумерованы, начиная с единицы, клетка с номером  $(1, 1)$  находится в левом верхнем углу.

Игрок может сделать несколько ходов по следующим правилам. Первым ходом он может указать на любой элемент таблицы  $(r_1, c_1)$ , и все числа в прямоугольнике  $(1, 1) - (r_1, c_1)$  меняют свой знак на противоположный. Здесь первое число обозначает номер строки, а второе — номер столбца. Следующие ходы игрока аналогичны, то есть выбирается прямоугольник  $(1, 1) - (r_i, c_i)$  и все числа опять меняют знак, однако каждый следующий прямоугольник должен полностью содержать предыдущий выбранный, но не совпадать с ним. Количество ходов при этом естественным образом ограничено размерами таблицы. При желании ходы можно и не выполнять совсем.

Например, если таблица состоит из 10 строк и 9 столбцов, то возможна в том числе такая последовательность ходов:  $(2, 2) - (4, 4) - (4, 6) - (5, 6)$ .

После того как все желаемые ходы выполнены, подсчитывается результат игры — сумма значений элементов преобразованной таблицы. Если сумма положительная, то игрок получает от заведения соответствующую сумму денег, а если отрицательная — платит ее.

Помогите игроку добиться наилучшего результата: выиграть как можно больше или хотя бы проиграть как можно меньше.

### Формат входных данных

В первой строке входного файла заданы два натуральных числа  $M$  и  $N$  ( $M \leq 1000$ ,  $N \leq 1000$ ). Каждая из следующих  $M$  строк содержит  $N$  целых чисел, по модулю не превышающих 1000, — элементы таблицы.

### Формат выходных данных

В первой строке выходного файла выведите максимальную сумму, которую можно получить с помощью оптимальной стратегии игры по описанным правилам. Во второй строке выведите целое число  $K$  — количество действий в алгоритме, дающем наилучший результат. В следующих  $K$  строках выведите последовательность из  $K$  пар чисел  $(r_i, c_i)$ , описывающих сам алгоритм.

Если дающих наилучший результат алгоритмов несколько, выведите любой.

### Примеры

стандартный ввод	стандартный вывод
2 3 3 6 4 1 -2 -7	21 2 1 3 2 3
4 4 2 -1 -5 -1 1 -3 -3 -1 0 -5 1 1 -1 -1 0 2	22 2 2 1 4 3

### Замечание