

Задача А. 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 В. 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 С. Доставка

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

Летом Костя решил постажироваться в Яндекс.Еде. К сожалению, его профессиональных навыков не хватило на то, чтобы его взяли в команду разработчиков, поэтому он стал развозить заказы.

В Костином городе ровно N домов, соединенных между собой $N - 1$ дорогами таким образом, что из каждого дома можно доехать до любого другого дома ровно одним способом. У каждой дороги есть своя длина D_i километров.

Для доставки Костя будет пользоваться своими электросамокатами. У каждого Костиного электросамоката есть фиксированная емкость аккумулятора T_i (будем считать, что емкость аккумулятора измеряется в количестве километров, которые можно проехать от одной зарядки). То есть, Костя проедет T_i километров без подзарядки, его самокат разрядится и не сможет ехать дальше.

Поскольку некоторые заказы нужно везти очень далеко, на всех Костиных самокатах есть солнечные батареи, с помощью которых можно подзарядить самокат. К сожалению, они работают только тогда, когда самокат стоит на месте, но зато они достаточно качественные, чтобы зарядить самокат за минуту. Каждый раз, когда Костя совершает остановку, он ждет, пока самокат зарядится до конца. Кроме вынужденных остановок, Костя останавливается у каждого дома, мимо которого он проезжает (в том числе, у первого и последнего), потому что соблюдает правила дорожного движения.

Косте за день приходит информация Q заказах на сегодня. Для каждого заказа скажите, какое минимальное время ему придется заряжать самокат.

Формат входных данных

Первая строка входного файла содержит числа N и Q — количество домов и путей соответственно ($1 \leq N, Q \leq 100\,000$). Следующие $N - 1$ строки содержат описание дорог. Каждая строка содержит три числа: U, V, D . U и V — это номера домов, соединенных дорогой, D — длина соответствующей дороги ($1 \leq U, V \leq N, 1 \leq D \leq 20\,000$)

Следующие Q строк описывают маршруты. В каждой из них вводится 3 числа: S, F, T ($1 \leq S, F \leq N, S \neq F$). S и F — это номер дома, от которого Костя начнет выполнение заказа, и номер дома, в который заказ надо привезти. T — это емкость аккумулятора самоката, на котором Костя будет доставлять этот заказ.

Формат выходных данных

Для каждого из Q заказов выведите наименьшее время, которое Косте придется провести в остановках.

Пример

стандартный ввод	стандартный вывод
7 5	7
1 2 1	9
2 3 2	5
2 4 3	5
4 5 4	12
4 6 5	
4 7 6	
3 7 2	
2 6 1	
5 7 3	
1 4 1	
3 7 1	

Задача D. Even Tree

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

Дан связный неориентированный взвешенный граф с n вершинами и m ребрами. Выберите в нем ровно $n - 1$ ребро так, чтобы граф все еще оставался связным и суммарный вес выбранных ребер был четным, либо скажите, что это невозможно.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 3 \cdot 10^5$; $n - 1 \leq m$) — количество вершин и ребер, соответственно.

Следующие m строк содержат информацию о ребрах. $(i + 1)$ -я строка входных данных содержит три целых числа v_i, u_i, w_i ($1 \leq v_i, u_i \leq n$; $0 \leq w_i \leq 10^9$) — начало и конец i -го ребра и его вес, соответственно. В графе могут быть кратные ребра и петли.

Формат выходных данных

В случае, если выбрать ребра так нельзя, выведите строку «NO».

Иначе выведите строку «YES» и номера ребер, которые следует выбрать.

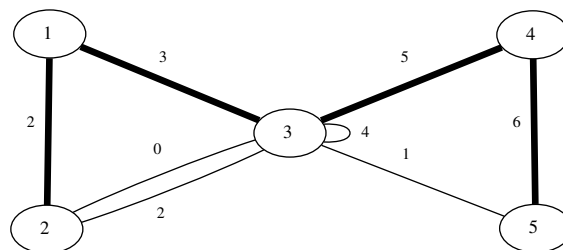
Если правильных ответов несколько, выведите любой из них.

Пример

стандартный ввод	стандартный вывод
5 8	YES
1 2 2	1 4 5 6
2 3 0	
2 3 2	
3 1 3	
3 4 5	
4 5 6	
5 3 1	
3 3 4	

Замечание

В первом тесте из примера мы выбрали ребра (1, 2), (3, 1), (3, 4) и (4, 5). Сам граф выглядит так:



Задача E. Два проспекта

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	8 секунд
Ограничение по памяти:	512 мегабайт

Для того чтобы сделать столицу Берляндии более привлекательным туристическим местом, великий король придумал следующий план: выбрать две улицы города и назвать их проспектами. Естественно, эти проспекты будут объявлены чрезвычайно важными историческими местами, что должно привлечь туристов со всего мира.

Столицу Берляндии можно представить в виде графа, вершинами которого являются перекрестки, а ребрами являются улицы, соединяющие два перекрестка напрямую. Всего в графе n вершин и m ребер, по любой улице можно двигаться в обоих направлениях, от любого перекрестка можно добраться до любого другого, перемещаясь только по улицам, каждая улица соединяет два различных перекрестка, и никакие две улицы не соединяют одинаковую пару перекрестков.

Чтобы снизить поток обычных горожан, перемещающихся по великим проспектам, было решено ввести платный проезд по каждому из них в обе стороны. Теперь за один проезд по проспекту нужно заплатить 1 тугрик. За проезд по остальным улицам платить не нужно.

Аналитики собрали выборку из k горожан, i -му из них надо ездить на работу от перекрестка a_i к перекрестку b_i . После выбора двух проспектов каждый горожанин будет добираться на работу вдоль пути, стоимость которого будет минимальна.

Для того чтобы заработать как можно больше денег, было решено выбрать в качестве двух проспектов две такие улицы, что суммарное количество тугриков, которые заплатят эти k горожан, будет максимально возможным. Помогите королю: по заданной схеме города и выборке горожан найдите, какие две улицы нужно сделать проспектами, и сколько тугриков заплатят горожане при таком выборе.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 10^5$) — количество наборов входных данных.

В первой строке описания каждого набора входных данных находятся два целых числа n и m ($3 \leq n \leq 500\,000$, $n - 1 \leq m \leq 500\,000$, $m \leq \frac{n(n-1)}{2}$) — количество перекрестков и улиц города.

В следующих m строках содержатся описания улиц, в i -й строке находятся два целых числа s_i и f_i ($1 \leq s_i, f_i \leq n$, $s_i \neq f_i$) — номера перекрестков, которые соединяет i -я улица. Гарантируется, что никакие две улицы не соединяют одну и ту же пару перекрестков, и что от любого перекрестка можно добраться до любого другого, перемещаясь только по улицам.

В следующей строке находится единственное целое число k ($1 \leq k \leq 500\,000$) — количество горожан в выборке.

В следующих k строках содержатся описания горожан, в i -й строке находятся два целых числа a_i и b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$) — i -й горожанин едет на работу от перекрестка a_i до перекрестка b_i .

Пусть M обозначает сумму значений m по всем наборам входных данных, а K означает сумму значений k по всем наборам входных данных. Гарантируется, что $M, K \leq 500\,000$.

Формат выходных данных

Для каждого набора входных данных в выведите ответ на задачу.

В первой строке ответа данных выведите суммарное количество тугриков, которое заплатят горожане.

Во второй строке ответа выведите два целых числа x_1 и y_1 — номера перекрёстков, дорогу между которыми нужно сделать первым проспектом.

В третьей строке ответа выведите два целых числа x_2 и y_2 — номера перекрёстков, дорогу между которыми нужно сделать вторым проспектом.

Номера перекрестков, соединенных улицей, можно выводить в произвольном порядке, каждая из двух выведенных улиц должна встречаться среди m улиц города, выбранные улицы должны быть различными.

Пример

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

Задача F. Спички детям не игрушка

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

Лена играет со спичками. Естественный вопрос, посещающий любого школьника, играющего со спичками — а можно ли поджечь спичкой дерево?

Скажем, что дерево — это связный граф без циклов, вершины которого пронумерованы целыми числами $1, 2, \dots, n$, в каждой вершине которого также записано некоторое целое число p_v , являющееся приоритетом вершины v . Все приоритеты различны.

Оказывается, что если поджечь дерево, то оно, как и можно было ожидать, сгорит целиком. Однако процесс этот не быстрый. Сначала у дерева сгорает лист (листом называется вершина, имеющая ровно одного соседа) с минимальным приоритетом, затем сгорает лист с минимальным приоритетом из оставшихся вершин дерева, и так далее. Таким образом, вершины превращаются в листья и сгорают до тех пор, пока от дерева не останется лишь одна вершина, после чего она тоже сгорает.

Лена приготовила дерево из n вершин и в каждой вершине записала приоритет $p_v = v$. Лене с одной стороны интересно посмотреть, как горит дерево, но с другой она понимает, что если дерево поджечь, оно исчезнет насовсем. Лена добрая девочка, и деревья ей жалко, так что она хочет ограничиться выяснением ответов на некоторые вопросы про процесс сгорания дерева в уме. Лена хочет ответить на q вопросов, каждый из которых относится к одному из трёх следующих видов:

1. «up v », присвоить вершине v приоритет $1 + \max(p_1, p_2, \dots, p_n)$
2. «when v », выяснить, какой по счёту сгорит вершина v , если дерево поджечь сейчас;
3. «compare v и u », выяснить, какая из вершин v и u сгорит раньше, если дерево поджечь сейчас.

Заметим, что если приоритеты всех вершин сейчас различны, то и после выполнения запроса «up» они тоже останутся различными. Исходно они различны, поэтому в любой момент времени порядок сгорания листьев определён однозначно.

Формат входных данных

Первая строка содержит два целых числа n и q ($2 \leq n \leq 200000$, $1 \leq q \leq 200000$) — количество вершин дерева и количество вопросов.

В i -й из следующих $n - 1$ строк находятся два целых числа v_i, u_i ($1 \leq v_i, u_i \leq n$), задающие концы i -го ребра дерева.

Каждая из оставшихся q строк содержит операцию одного из трёх типов.

1. «up v » ($1 \leq v \leq n$) — присвоить новый приоритет вершине v
2. «when v » ($1 \leq v \leq n$) — определить момент сгорания вершины v для текущего дерева
3. «compare v и u » ($1 \leq v, u \leq n, v \neq u$) — определить, какая из вершин v и u сгорит раньше для текущего дерева.

Гарантируется, что среди запросов хотя бы один имеет тип «when» или «compare».

Формат выходных данных

Для каждого запроса типа «when» нужно вывести одно целое число от 1 до n — момент времени, когда сгорит вершина v .

Для запроса типа «compare» выведите v или u , в зависимости от того, какая вершина сгорит раньше.

Примеры

стандартный ввод	стандартный вывод
5 7 1 5 1 2 1 3 4 3 when 1 when 2 when 3 when 4 when 5 compare 2 3 compare 3 4	4 1 3 2 5 2 4
5 5 1 5 1 2 1 3 4 3 up 1 compare 2 4 compare 4 3 compare 3 1 compare 1 5	2 4 3 5