

## Задача А. Праздник к нам приходит

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

В уездном городе Т все заняты подготовкой к новому году. Таня вот, например, готовится вручать подарки в новогоднюю ночь (в городе Т именно Таня исполняет роль Деда Мороза).

В городе Т  $n$  домов, некоторые из которых соединены улицами, причем так, что между любыми двумя домами есть ровно один путь.

Таня уже заготовила  $m$  мешков с подарками (в этом году она дарит серые футболки с желтым единорогом), но столкнулась с проблемой. Ей ведь самой тоже нужно найти дом в городе Т для празднования Нового Года. Раздачей подарков же, как обычно, будут заниматься олениа. Одному олененку можно дать ровно один мешок и отправить его в путь. При этом олениа не ходят по одной и той же улице дважды.  $i$ -й мешок предназначен для жителей всех домов на пути от  $a_i$  до  $b_i$ . Поэтому считается, что олененку можно дать  $i$ -й мешок с подарками, если он сможет выйти из места празднования Тани и пройти через все дома на пути от  $a_i$  до  $b_i$ , при этом не проходя через одну и ту же улицу дважды.

Помогите Тани найти дом для празднования так, чтобы она смогла отправить как можно больше мешков с подарками. В данной задаче можно считать, что количество олениа не ограничено.

### Формат входных данных

В первой строке задано число  $n$  — количество домов в городе Т ( $2 \leq n \leq 2 \cdot 10^5$ ).

В следующих  $n - 1$  строках описаны улицы. Улица задаётся числами  $x_i$  и  $y_i$  — номерами домов, которые она соединяет ( $1 \leq x_i, y_i \leq n$ ,  $x_i \neq y_i$ ). Гарантируется, что между любыми двумя домами существует единственный путь.

В следующей строке задано число  $m$  — количество мешков у Тани ( $1 \leq m \leq 2 \cdot 10^5$ ).

В следующих  $m$  строках описаны мешки с футболками. В  $i$ -й из них заданы числа  $a_i$  и  $b_i$  — начало и конец  $i$ -го пути ( $1 \leq a_i, b_i \leq n$ ,  $a_i \neq b_i$ ). Пути могут пересекаться и совпадать.

Дома нумеруются с единицы.

### Формат выходных данных

Выведите одно число — максимальное число мешков с подарками, которые Таня сможет раздать, если выберет оптимальный дом для празднования.

### Пример

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

## Задача В. Самое дешевое ребро

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

Дано подвешенное дерево с корнем в первой вершине. Все ребра имеют веса (стоимости). Вам нужно ответить на  $M$  запросов вида “найти у двух вершин минимум среди стоимостей ребер пути между ними”.

### Формат входных данных

В первой строке файла записано одно число —  $n$  (количество вершин).

В следующих  $n - 1$  строках записаны два числа —  $x$  и  $y$ . Число  $x$  на строке  $i$  означает, что  $x$  — предок вершины  $i$ ,  $y$  означает стоимость ребра.

$x < i$ ,  $|y| \leq 10^6$ .

Далее  $m$  запросов вида  $(x, y)$  — найти минимум на пути из  $x$  в  $y$  ( $x \neq y$ ).

Ограничения:  $2 \leq n \leq 5 \cdot 10^4$ ,  $0 \leq m \leq 5 \cdot 10^4$ .

### Формат выходных данных

Выведите  $m$  ответов на запросы.

### Пример

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

## Задача С. 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  $1 \leq N \leq 500\,000$ ,  $K = M/2$ ,  $0 \leq M \leq 1\,000\,000$ .

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 D. 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.

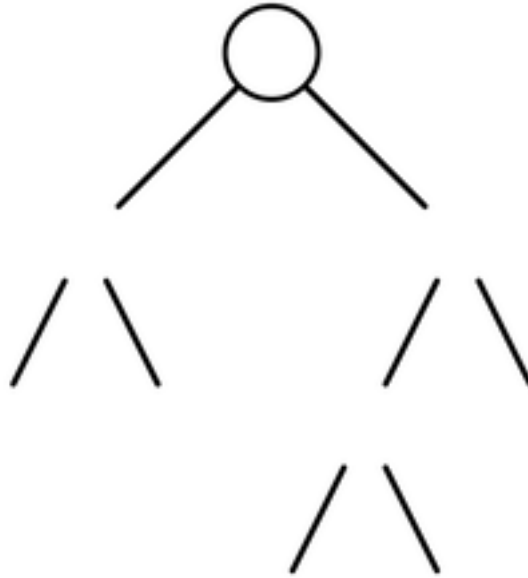
### Примеры

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

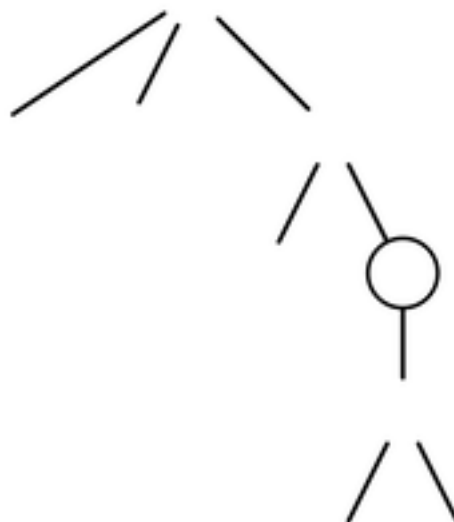
## Задача E. Dynamic LCA

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 F. RMQ

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

Есть массив из  $N$  целых чисел и  $M$  запросов вида: найдите минимум на отрезке с концами  $l_i, r_i$ .

### Формат входных данных

Входной файл содержит  $T$  наборов тестовых данных. Каждый набор тестовых данных задаётся числами  $N, M, A, B$  ( $1 \leq N \leq 25\,000, 1 \leq A, B \leq 10^9$ ), где  $N$  — размер массива,  $M$  — число запросов.

Массив и запросы нужно получить следующим образом: выпишем последовательность чисел  $C_i = (A \cdot i + B) \bmod 2^{32}$ .

Элементы последовательности с номерами от 1 до  $N$  — элементы массива. Элементы последовательности с номерами от  $N + 1$  до  $N + 2 \cdot M$  взятые по модулю  $N$  образуют  $M$  пар чисел, которые являются границами отрезков запросов. Ввод заканчивается числами 0 0 0 0. Массив индексируется с нуля.

Сумма  $N$  по всем наборам тестовых данных не превосходит  $10^8$ . Сумма  $M$  по всем наборам тестовых данных не превосходит  $2 \cdot 10^7$ .

### Формат выходных данных

Для каждого набора тестовых данных выведите сумму по всем запросам.

### Пример

стандартный ввод	стандартный вывод
10 10 955379886 619166003	7671393960
0 0 0 0	

### Замечание

Массив:

1574545889 2529925775 3485305661 145718251 1101098137 2056478023 3011857909  
3967237795 627650385 1583030271

Запросы:

7 3  
3 9  
5 1  
7 7  
3 9  
5 5  
1 7  
3 9  
9 5  
1 7

## Задача G. 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 Н. 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 I. 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

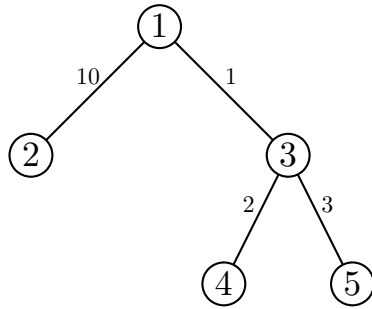
Выведите  $q$  чисел — ответы на запросы в порядке, в котором они заданы во входных данных.

## 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 J. Спички детям не игрушка

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