

Задача А. LCA Problem

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

Задано подвешенное дерево, содержащее n ($1 \leq n \leq 10^5$) вершин, пронумерованных от 0 до $n - 1$. Требуется ответить на m ($1 \leq m \leq 10^6$) запросов о наименьшем общем предке для пары вершин.

Формат входных данных

Первая строка содержит два числа: n и m . Корень дерева имеет номер 0.

Вторая строка содержит $n - 1$ целых чисел, i -е из этих чисел равно номеру родителя вершины i .

Следующие m строк содержат два целых числа от 0 до $n - 1$: a_i, b_i — вершины из i -го запроса.

Формат выходных данных

Выведите в выходной файл сумму номеров вершин — ответов на все запросы.

Примеры

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

Задача B. LCA Problem Revisited

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

Задано подвешенное дерево, содержащее n ($1 \leq n \leq 10^5$) вершин, пронумерованных от 0 до $n - 1$. Требуется ответить на m ($1 \leq m \leq 10^7$) запросов о наименьшем общем предке для пары вершин.

Запросы генерируются следующим образом. Заданы числа a_1, a_2 и числа x, y, 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 (a_1, a_2) . Если ответ на $i - 1$ -й запрос равен v , то i -й запрос имеет вид $((a_{2i-1} + v) \bmod n, a_{2i})$.

Формат входных данных

Первая строка содержит два числа: n и m . Корень дерева имеет номер 0.

Вторая строка содержит $n - 1$ целых чисел, i -е из этих чисел равно номеру родителя вершины i .

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, z , эти числа неотрицательны и не превосходят 10^9 .

Формат выходных данных

Выведите в выходной файл сумму номеров вершин — ответов на все запросы.

Примеры

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

Задача С. Учиться! - EASY

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	64 мегабайта

Каждый год огненное 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.

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 ($1 \leq m \leq 100\,000$). 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. Следующие m строк содержит два целых числа в диапазоне от 0 до $n - 1$: a_i и b_i .

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	1
1 2 0 0 0 0	0

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, сдавших 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 дорог, 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 Е. Дуумвират 2

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

Надо бы всё-таки написать нормальную легенду, а то как-то не очень. И без легенды непонятно, почему задача так называется

Но пока легенды нет, вот формальное условие:

Вам дано дерево на n вершинах. В вершинах записаны числа. Требуется отвечать на запросы двух видов:

- $? v u$ — узнать сумму значений чисел, записанных в вершинах на пути из v в u .
- $! v x$ — сделать значение, записанное в вершине v равным x .

Формат входных данных

В первой строке записано число n — количество вершин дерева ($1 \leq n \leq 10^6$). Во второй строке записаны через пробел n чисел v_i ($|v_i| < 10^9$), задающие значения в вершинах. В следующих $n - 1$ строках описаны ребра дерева. В $(i + 2)$ -й строке записаны номера вершин a_i, b_i ($1 \leq a_i, b_i \leq n$), означающие, что в дереве есть ребро из вершины a_i в вершину b_i .

Далее на отдельной строке записано число m — количество запросов ($1 \leq m \leq 10^6$). После этого идут m строк с описанием запросов, в очередной строке может быть написано $?vu$ — узнать сумму на пути из v в u ($1 \leq v, u \leq n$). Или $!vx$ — изменить значение в вершине v на x ($1 \leq v \leq n$, $-10^9 \leq x \leq 10^9$).

Формат выходных данных

Для каждого запроса виде $?vu$ выведите искомую величину.

Пример

стандартный ввод	стандартный вывод
7	0
0 -7 -2 -7 1 4 8	-5
3 6	-2
7 6	-8
1 3	
5 1	
4 6	
2 1	
7	
? 1 1	
? 2 6	
! 2 -8	
! 7 -6	
! 4 -6	
? 1 3	
? 1 2	

Задача F. Очередные запросы на дереве

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 256 мегабайт

Вам дано дерево, состоящее из n вершин. Каждое ребро дерева имеет длину, которая является некоторым натуральным числом. Вам нужно ответить на q запросов, каждый задается парой вершин a и b и числом x . Ответом на запрос является количество ребер на пути от a до b , длина которых $\leq x$.

Формат входных данных

В первой строке находится два целых числа n и q ($2 \leq n, q \leq 200\,000$), разделенных пробелом. В следующих $n - 1$ строках находится по три целых числа s, f, l ($1 \leq s, f \leq n, 1 \leq l \leq 10^6$), разделенных пробелами, обозначающих ребро (s, f) длины l . В следующих q строках находится по три целых числа a, b, x ($1 \leq a, b \leq n, 1 \leq x \leq 10^6, a \neq b$), разделенных пробелами, обозначающих запрос a, b, x .

Формат выходных данных

Выведите q строк, в i -й строке ответ на i -й запрос.

Пример

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

Задача G. Повогодняя ёлка

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

Лекция по ДО прошла, но Саша решил не убирать своё любимое дерево (но уже обычно), а немного преукрасить его. К нему пришли его друзья Аня и Ваня, чтобы помочь преукрасить дерево.

Дерево подвешено за вершину 1.

Вам требуется обрабатывать запросы двух типов:

1. Перекрасить все вершины из поддерева вершины v в цвет c .
2. Найти количество различных цветов в поддерева вершины v .

Формат входных данных

В первой строке находится пара целых чисел n, m ($1 \leq n, m \leq 4 \cdot 10^5$) — количество вершин в дереве и количество запросов.

Во второй строке находятся n целых чисел c_i ($1 \leq c_i \leq 60$) — цвет i -й вершины.

В каждой из следующих $n - 1$ строк находится пара целых чисел x_j, y_j ($1 \leq x_j, y_j \leq n$) — вершины, которые соединяет j -е ребро дерева. Гарантируется, что вам задано корректное неориентированное дерево.

В последних m строках находятся описания запросов. Каждое описание начинается с целого числа t_k ($1 \leq t_k \leq 2$) — типа запроса.

- Для запросов первого типа далее следует пара целых чисел v_k, c_k ($1 \leq v_k \leq n, 1 \leq c_k \leq 60$) — номер вершины дерева, в поддерева которой все вершины перекрашиваются в цвет c_k . - Для запросов второго типа далее следует целое число v_k ($1 \leq v_k \leq n$) — номер вершины, в поддерева которой нужно посчитать количество различных цветов.

Формат выходных данных

Для каждого запроса второго типа выведите одно целое число a — количество различных цветов в поддерева вершины, заданной в запросе.

Числа нужно выводить в отдельных строках, в порядке появления запросов во входных данных.

Пример

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

Задача Н. Вирусы и антивирусы

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

Антивирусная IT-компания имеет официальную иерархическую структуру управления. В ней есть босс — единственный сотрудник, над которым нет начальника. Каждый из остальных сотрудников подчинён ровно одному сотруднику — своему начальнику. Начальник может иметь нескольких подчинённых и отдавать или передавать приказы любому из них. Приказы могут передаваться от одного сотрудника другому только по цепочке, каждый раз от начальника к его подчинённому. Сотрудник А главнее сотрудника Б в этой иерархии, если А может отдать или передать приказ сотруднику Б непосредственно, или через цепочку подчинённых. Босс главнее любого сотрудника.

Оказалось, что все сотрудники объединены ещё в одну организованную подобным образом тайную иерархическую структуру, производящую компьютерные вирусы. В тайной структуре может быть другой босс, а у сотрудников — другие начальники.

Будем называть пару сотрудников А и Б устойчивой, если А главнее Б и в основной, и в тайной иерархических структурах.

Требуется написать программу, определяющую количество устойчивых пар в компании.

Формат входных данных

В первой строке задано число N — количество сотрудников компании ($1 \leq N \leq 100\,000$).

Во второй строке — N целых чисел a_i , где $a_i = 0$, если в официальной иерархии сотрудник с номером i является боссом, в противном случае a_i равно номеру непосредственного начальника сотрудника номер i .

В третьей строке — N целых чисел b_i , где $b_i = 0$, если в тайной иерархии сотрудник с номером i является боссом, в противном случае b_i равно номеру непосредственного начальника сотрудника номер i .

Нумерация сотрудников ведется с единицы в том порядке, в каком они упомянуты во входном файле.

Формат выходных данных

Выходной файл должен содержать единственное число — количество устойчивых пар.

Примеры

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

Задача I. Антивещество

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	128 мегабайт

Компания тестирует технологию получения антивещества, используемого в качестве топлива в межпланетном звездолёте. Антивещество получается в результате специальных экспериментов в реакторе.

Известно n типов экспериментов, приводящих к получению антивещества. В результате проведения эксперимента i -го типа в выходной контейнер реактора добавляется от l_i до r_i граммов антивещества. Из соображений безопасности запрещается накапливать в контейнере более a граммов антивещества.

Затраты на проведение эксперимента i -го типа составляют c_i , а стоимость одного грамма полученного антивещества составляет 10^9 .

Если после проведения экспериментов в контейнере образовалось t граммов антивещества, а суммарные затраты на проведение экспериментов в реакторе составили s , то прибыль определяется по формуле $(t \cdot 10^9 - s)$. Компании необходимо разработать стратегию проведения экспериментов, позволяющую максимизировать прибыль, которую можно гарантированно получить.

В зависимости от результатов предыдущих экспериментов стратегия определяет, эксперимент какого типа следует провести, или решает прекратить дальнейшее выполнение экспериментов.

Стратегия позволяет гарантированно получить прибыль x , если при любых результатах проведения экспериментов: во-первых, в контейнере реактора оказывается не более a граммов антивещества, во-вторых, прибыль составит не менее x . Например, пусть возможен только один тип эксперимента, порождающий от 4 до 6 граммов антивещества, затраты на его проведение равны 10, а вместимость контейнера составляет 17 граммов. Тогда после двукратного проведения эксперимента в контейнере может оказаться от 8 до 12 граммов антивещества. Если получилось 12 граммов, то больше проводить эксперимент нельзя, так как в случае получения 6 граммов антивещества контейнер может переполниться. В остальных случаях можно провести эксперимент в третий раз и получить от 12 до 17 граммов антивещества. В худшем случае придётся провести эксперимент трижды, затратив в сумме 30, прибыль составит $(12 \cdot 10^9 - 30) = 11999999970$.

Требуется написать программу, которая определяет максимальную прибыль x , которую гарантированно можно получить.

Формат входных данных

Первая строка входных данных содержит два целых числа: n — количество типов экспериментов и a — максимально допустимое количество антивещества в контейнере ($1 \leq n \leq 100, 1 \leq a \leq 2000000$).

Следующие n строк содержат по три целых числа l_i, r_i, c_i — минимальное и максимальное количество антивещества, получаемое в результате эксперимента типа i , и затраты на эксперимент этого типа, соответственно ($1 \leq l_i \leq r_i \leq a, 1 \leq c_i \leq 100$).

Формат выходных данных

Выходные данные должны содержать одно целое число — максимальную прибыль x , которую гарантированно можно получить.

Примеры

стандартный ввод	стандартный вывод
1 17 4 6 10	11999999970
2 11 2 2 100 3 5 5	9999999890
1 1 1 1 1	999999999

Задача J. Прямоугольники

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

Когда-то тут была легенда про вёдра, но её съели.

Жюри олимпиады

Есть таблица T размера $N \times M$. Элементами таблицы являются прямоугольники T_{ij} , где $0 \leq i < N$ и $0 \leq j < M$. Прямоугольник T_{ij} задаётся четвёркой чисел $(x_1^{ij}, y_1^{ij}, x_2^{ij}, y_2^{ij})$, где (x_1^{ij}, y_1^{ij}) и (x_2^{ij}, y_2^{ij}) — координаты противоположных углов прямоугольника. Стороны прямоугольника параллельны осям координат.

Далее вам поступают запросы. Каждый запрос состоит из четырёх чисел: (r_1, c_1, r_2, c_2) . Ответом на такой запрос является площадь фигуры, являющейся пересечением всех прямоугольников T_{ij} таких, что $\min(r_1, r_2) \leq i \leq \max(r_1, r_2)$ и $\min(c_1, c_2) \leq j \leq \max(c_1, c_2)$. Запросов очень много, поэтому мы просим вас вывести сумму ответов на все запросы по модулю $10^9 + 7$.

Формат входных данных

В первой строке записаны два целых числа N и M — размеры таблицы T ($1 \leq N, M \leq 127$). Далее в N строках описывается таблица T : в $(i + 1)$ -й строке $(j + 1)$ -я четвёрка чисел $x_1^{ij} y_1^{ij} x_2^{ij} y_2^{ij}$ описывает прямоугольник T_{ij} . Гарантируется, что $|x_k^{ij}|, |y_k^{ij}| \leq 10^6$.

Дальше в отдельной строке записано четыре числа. Первое из них, число Q — количество запросов ($1 \leq Q \leq 5 \cdot 10^6$). Следующие три числа — это A, B, v_0 ($0 \leq A, B, v_0 < 10^9 + 7$). При помощи этих чисел генерируется бесконечная последовательность $\{v_i\}$ по правилу $v_i = (A \cdot v_{i-1} + B) \bmod (10^9 + 7)$.

После этого k -й запрос (запросы нумеруются с единицы) задаётся следующей четвёркой чисел: $(v_{4k-3} \bmod N, v_{4k-2} \bmod M, v_{4k-1} \bmod N, v_{4k} \bmod M)$.

Формат выходных данных

Выведите сумму ответов на все запросы по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
2 2 0 0 2 2 1 1 3 3 0 3 2 1 1 2 3 0 1 500000003 4 2	1
3 2 8 -1 -7 6 6 8 9 10 -4 -10 4 9 -3 -8 6 9 -2 -9 3 8 -5 7 7 3 5 303164476 273973578 65779139	85

Замечание

В первом примере запрос имеет вид $(1, 0, 0, 1)$, то есть это запрос ко всей таблице. Пересечением всех прямоугольников является квадрат с углами в точках $(1, 1)$ и $(2, 2)$. Его площадь равна 1.

Во втором примере запросы имеют вид $(0, 1, 1, 1)$, $(1, 0, 2, 0)$, $(0, 0, 2, 1)$, $(0, 1, 1, 1)$, $(0, 1, 0, 0)$. На второй запрос ответ — 85, на остальные — 0.

Задача К. Очередная попарная сумма

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

Вам дано неориентированное дерево на N вершинах. i -е ребро соединяет вершины v_i и u_i . Также у вас есть последовательность чисел $A = (A_1, \dots, A_N)$.

Обозначим за $f(i, j)$ следующую величину: если $A_i = A_j$, то $f(i, j)$ равно количеству рёбер на кратчайшем пути в дереве от i до j ; если же $A_i \neq A_j$, то $f(i, j) = 0$.

Посчитайте и выведите величину $\sum_{i=1}^{N-1} \sum_{j=i+1}^N f(i, j)$

Формат входных данных

В первой строке находится одно целое число N ($1 \leq N \leq 2 \cdot 10^5$).

В i -й из следующих $N - 1$ строк содержатся два целых числа u_i и v_i — концы i -го ребра ($1 \leq v_i, u_i \leq N$).

В следующей строке дана последовательность A из N чисел, разделённых пробелом ($1 \leq A_i \leq N$).

Гарантируется, что рёбра образуют дерево.

Формат выходных данных

Выведите требуемую величину.

Примеры

стандартный ввод	стандартный вывод
4 3 4 4 2 1 2 2 1 1 2	4
8 8 6 3 8 1 4 7 8 4 5 3 4 8 2 1 2 2 2 3 1 1 3	19

Задача L. Туризм

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

Королевство JOI состоит из n островов, пронумерованных от 1 до n , и $n - 1$ мостов. Мост i соединяет острова a_i и b_i в обоих направлениях. С любого острова можно добраться до любого другого.

В королевстве есть m достопримечательностей. Достопримечательность j находится на острове c_j . На одном острове может находиться несколько достопримечательностей.

В королевство прибыли q путешественников. Путешественник k хочет посетить все достопримечательности с номерами от l_k до r_k включительно. Он может начать и закончить путешествие на любых островах и перемещаться по мостам в произвольном порядке.

Отрезки $[l_1, r_1], [l_2, r_2], \dots, [l_q, r_q]$ последовательно разбивают все достопримечательности: $l_1 = 1$, $l_{k+1} = r_k + 1$ для всех $1 \leq k < q$ и $r_q = m$.

Для каждого путешественника определите минимальное количество различных островов, которые ему придётся посетить. Начальный и конечный острова также учитываются. Каждый остров учитывается только один раз, независимо от количества посещений.

Формат входных данных

Первая строка содержит три целых числа n , m и q ($1 \leq n, m, q \leq 100\,000$).

Следующие $n - 1$ строк содержат по два целых числа a_i и b_i ($1 \leq a_i, b_i \leq n$) — концы моста i . Гарантируется, что мосты образуют дерево.

Следующая строка содержит m целых чисел $c_1, c_2, \dots, c_m$ ($1 \leq c_j \leq n$) — острова, на которых находятся достопримечательности.

Следующие q строк содержат по два целых числа l_k и r_k ($1 \leq l_k \leq r_k \leq m$). Гарантируется, что $l_1 = 1$, $l_{k+1} = r_k + 1$ для всех $1 \leq k < q$ и $r_q = m$.

Формат выходных данных

Выведите q строк. В строке k выведите минимальное количество различных островов, которые необходимо посетить путешественнику k .

Пример

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

Задача М. Цвета в дереве

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

Задано подвешенное дерево, содержащее n ($1 \leq n \leq 1\,000\,000$) вершин. Каждая вершина покрашена в один из n цветов. Требуется для каждой вершины v вычислить количество различных цветов, встречающихся в поддереве с корнем v .

Формат входных данных

В первой строке входного файла задано число n .

Следующие n строк описывают вершины, по одной в строке. Описание очередной вершины i имеет вид $p_i\ c_i$, где p_i — номер родителя вершины i , а c_i — цвет вершины i ($1 \leq c_i \leq n$). Для корня дерева $p_i = 0$.

Формат выходных данных

Выведите n чисел, обозначающих количество различных цветов в поддеревьях с корнями в вершинах $1, \dots, n$.

Пример

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

Задача N. Задача без легенды

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

Дано дерево на n вершинах. Вам нужно обрабатывать запросы двух типов:

1. Дана вершина v и число x . Добавьте x ко всем вершинам в поддереве v .
2. Даны вершины u, v . Найдите сумму чисел в вершинах на пути из u в v .

Формат входных данных

В первой строке ввода находятся числа n, q . $1 \leq n, q \leq 10^5$.

В следующих $(n - 1)$ строках находятся ребра дерева, заданные парами (u_i, v_i) . $1 \leq u_i, v_i \leq n$.

В следующих q строках находятся запросы:

- если текущий запрос первого типа, то строка имеет вид «+ v x». $1 \leq x \leq 10^9$
- иначе строка имеет вид «? u v», и вам нужно найти сумму на пути из u в v по модулю $10^9 + 9$.

Формат выходных данных

Введите q чисел — ответы на запросы.

Пример

стандартный ввод	стандартный вывод
6 7	18
1 2	28
1 3	21
2 4	
2 5	
3 6	
+ 2 4	
+ 1 2	
? 4 6	
+ 3 5	
? 5 6	
+ 4 3	
? 4 5	