

Задача А. Гремучая ива

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

Гремучей Иве n лет. На первом году жизни у неё был только *корень*. Далее каждый год Ива отращивала себе *ветку* длиной ровно один фут, которая оканчивается *листом*. Корень тоже считается листом.

Гарри Поттер и Рон Уизли очень часто нужно проникать в Визжащую Хижину, которую охраняет Ива. Они решили подойти основательно и тщательно исследовать строение Гремучих Ив. Они просят вас найти историю изменения *диаметров* Ивы. Диаметр Ивы называется максимальное расстояние в футах между двумя листьями дерева.

Формат входных данных

Первая строке содержит целое число n — возраст ивы ($1 \leq n \leq 10^6$).

Следующие n строк содержат описание её листьев. Каждая строка содержит номер листа p_i , который является её родителем ($1 \leq p_i \leq i$). Корень имеет номер 1.

Формат выходных данных

Выведите диаметр после добавления каждого листа.

Примеры

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

Замечание

Задача В. Максимумы

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 6 секунд
Ограничение по памяти: 256 мегабайт

К сожалению, у Дани не хватило времени, чтобы написать нескучное условие к этой задаче.

Вам дан массив a_0, \dots, a_{n-1} , который задаётся числами n, a_0, b, c, d следующим образом:

$$b_i = (b_{i-1} \cdot c + d) \bmod 2^{31} \text{ для } i \geq 1$$

$$a_i = (a_{i-1} + 1 - 2 \cdot ((b_i \bmod 239179) \bmod 2)) \text{ для } i \geq 1$$

Обратите внимание, что два соседних числа отличаются либо на $+1$, либо на -1 .

Ответьте на n запросов, i -й запрос — максимум на отрезке $[\min(l_i, r_i), \max(l_i, r_i)]$ для $i = 0 \dots n-1$.

Пусть ans_i — ответ на i -й запрос. Будем считать, что $ans_{-1} = 0$. Вам задаётся число x_0 . Далее, l_i и r_i, x_i вычисляются так:

$$l_i = (x_i + ans_{i-1}) \bmod n \text{ для } i \geq 0$$

$$r_i = (l_i + i) \bmod n \text{ для } i \geq 0$$

$$x_i = (x_{i-1} \cdot 1103515245 + 12345) \bmod 2^{31} \text{ для } i \geq 1$$

При взятии по модулю обратите внимание на то, что ans_i бывают отрицательными.

Формат входных данных

Единственная строка ввода содержит шесть целых чисел n, a_0, b, c, d, x_0 .

- $1 \leq n \leq 2 \cdot 10^7$
- $-10^9 \leq a_0 \leq 10^9$
- $0 \leq b, c, d, x_0 \leq 2^{31} - 1$

Формат выходных данных

Выведите сумму ответов на все запросы.

Примеры

стандартный ввод	стандартный вывод
4 1 2 3 4 0	11
100500 -1 23 45 67 89	-8614564

Задача С. Максимумы возвращаются

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 6 секунд
Ограничение по памяти: 512 мегабайт

Вам дан массив a_0, \dots, a_{n-1} , который задаётся числами n, a_0, b_0, c, d следующим образом:

$$b_i = (b_{i-1} \cdot c + d) \bmod 2^{31} \text{ для } i \geq 1$$

$$a_i = (a_{i-1} + 1 - 2 \cdot ((b_i \bmod 239179) \bmod 2)) \text{ для } i \geq 1$$

Обратите внимание, что два соседних числа отличаются либо на $+1$, либо на -1 .

Ответьте на n запросов, i -й запрос — поиск количества максимумов на отрезке $[\min(l_i, r_i), \max(l_i, r_i)]$ для $i = 0 \dots n - 1$. Пусть ans_i — ответ на i -й запрос. Будем считать, что $ans_{-1} = 0$. Вам задаётся число x_0 . Далее, l_i и r_i, x_i вычисляются так:

$$l_i = (x_i + ans_{i-1}) \bmod n \text{ для } i \geq 0$$

$$r_i = (l + i) \bmod n \text{ для } i \geq 0$$

$$x_i = (x_{i-1} \cdot 1103515245 + 12345) \bmod 2^{31} \text{ для } i \geq 1$$

Формат входных данных

Единственная строка ввода содержит шесть целых чисел n, a_0, b, c, d, x_0 .

- $1 \leq n \leq 2 \cdot 10^7$
- $-10^9 \leq a_0 \leq 10^9$
- $0 \leq b, c, d, x_0 \leq 2^{31} - 1$

Формат выходных данных

Выведите сумму ответов на все запросы.

Примеры

стандартный ввод	стандартный вывод
4 1 2 3 4 0	4
100500 -1 23 45 67 89	173287

Задача D. 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 E. Чип и Дейл в лабиринте

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

Чип и Дейл спешат на помощь! Но внимательные зрители знают, что помощь как правило нужна самим Чипу и Дейлу, поэтому сегодня вам надо будет сыграть роль сообразительной Гаечки. Итак, Чип и Дейл снова попали в лапы к Толстопузу. Кот очень не любит грызунов и поэтому приготовил им изощренное испытание. Он собирается поместить их в лабиринт и посмотреть смогут ли они из него выбраться. Лабиринт представляет собой дерево, в котором каждое ребро имеет одно направление. Гаечка подслушала разговор Толстопузу со своими сообщниками и теперь знает несколько возможных вариантов: в какую точку лабиринта поместят её друзей, и где будет выход. Для каждого такого варианта она хочет понять, смогут ли Чип и Дейл найти выход, или нет.

Формат входных данных

В первой строке входного файла записано число n ($1 \leq n \leq 10^5$) — число вершин в дереве. В следующих $n - 1$ строках описаны ребра дерева. В $i + 1$ строке файла записаны два числа a_i, b_i ($1 \leq a_i, b_i \leq n$), означающие, что существует ребро из a_i в b_i .

Далее записано число m ($1 \leq m \leq 10^5$) — число запросов. После этого идет описание запросов, каждый запрос в новой строке. Для каждого запроса задается x_i, y_i ($1 \leq x_i, y_i \leq n$) — точка, в которую поместят Чипа и Дейла, и выход из лабиринта соответственно.

Формат выходных данных

Для каждого запроса надо в отдельной строке вывести Yes, если бурундуки смогут найти выход, и No иначе.

Пример

стандартный ввод	стандартный вывод
4	Yes
1 2	Yes
3 1	No
4 1	Yes
6	No
1 2	No
3 2	
2 3	
4 2	
4 3	
2 1	

Задача F. В бухгалтерии опять всё перепутали

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
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 Лулу 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. 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 Н. 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 I. Дерево

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

Дано дерево из n вершин и q запросов.

Каждый запрос начинается с трех целых чисел k , m и r , и продолжается k вершинами дерева a_1, a_2, \dots, a_k . Чтобы ответить на запрос, предположите, что дерево подвешено за вершину r . Рассмотрим разбиения данных k вершин на **не более чем** m групп так, что выполняются следующие условия:

- Каждая вершина принадлежит ровно одной группе, каждая группа содержит хотя бы одну вершину.
- Ни в одной группе нет двух вершин таких, что одна является предком (не обязательно непосредственным) другой.

Выведите количество различных таких разбиений по модулю $10^9 + 7$ для каждого запроса.

Формат входных данных

Первая строка содержит два целых числа n и q ($1 \leq n, q \leq 10^5$) — количество вершин в дереве и количество запросов, соответственно.

Каждая из следующих $n - 1$ вершин содержит два целых числа u и v ($1 \leq u, v \leq n, u \neq v$), обозначающие ребро между вершинами u и v . Гарантируется, что данный граф является деревом.

Каждая из следующих q строк начинается с трех целых чисел k , m и r ($1 \leq k, r \leq n, 1 \leq m \leq \min(300, k)$) — количество вершин, максимальный размер группы и корень дерева для данного запроса, соответственно. После этого следуют k различных целых чисел a_1, a_2, \dots, a_k ($1 \leq a_i \leq n$) — вершины текущего запроса.

Гарантируется, что сумма значений k по всем запросам не превосходит 10^5 .

Формат выходных данных

Выведите q строк, где i -я строка содержит ответ на i -й запрос.

Примеры

стандартный ввод	стандартный вывод
<pre> 7 2 5 4 2 6 5 3 1 2 7 5 4 6 3 3 2 7 4 3 3 1 4 6 2 1 </pre>	<pre> 2 0 </pre>
<pre> 7 2 4 7 2 5 4 1 5 1 5 6 4 3 3 3 2 7 1 4 2 1 6 3 2 </pre>	<pre> 1 1 </pre>
<pre> 5 2 3 5 4 5 4 2 1 4 2 2 3 1 2 2 2 4 5 4 </pre>	<pre> 2 1 </pre>

Замечание

Рассмотрим первый пример.

В первом запросе нужно разделить три данные вершины (7, 4 и 3) на не более чем три группы, считая, что корнем дерева является вершина 2. Когда дерево подвешено за вершину 2, вершина 4 является предком вершин 3 и 7. Поэтому нельзя все вершины отнести к одной группе. Есть только 1 способ разделить эти вершины на две группы: [4] и [3, 7]. Кроме того, есть один способ разделить данные вершины на три группы: [7], [4] и [3]. Таким образом, есть всего 2 способа разбить данные вершины на не более чем три группы.

Во втором запросе дерево подвешено за вершину 4, при этом 6 является предком 2, а 2 является предком 1. Поэтому нельзя все вершины отнести к одной группе.

Задача J. Джон и волшебная коробка

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

Это интерактивная задача.

У Джона есть массив из n загадочных чисел. Ещё у него есть волшебная коробка, которая может сделать из двух чисел одно (исходные числа остаются на месте). Пусть $x \circ y$ — результат работы коробки с числами x и y . После долгих экспериментов Джон обнаружил два свойства коробки:

- $x \circ y = y \circ x$
- $x \circ (y \circ z) = (x \circ y) \circ z$

У Джона есть q племянниц, и каждая из них любит все числа из массива Джона, кроме некоторых k из них. Джон хочет подарить каждой племяннице результат комбинации с помощью коробки всех чисел из его массива, кроме тех k , которые эта племянница не любит.

Джон любит своих племянниц, но из-за многочисленных экспериментов волшебная коробка изнасилась. Джон не сможет использовать её больше, чем $4(n + q + q \cdot k)$ раз. Помогите ему получить все требуемые комбинации!

Протокол взаимодействия

В начале работы решение получает строку с тремя целыми числами n , q и k ($1 \leq q \leq n \leq 2000$, $n \geq 3$, $2 \leq 2^k \leq n$). Далее следует строка, содержащая n целых чисел a_1, \dots, a_n ($0 \leq a_i < 2^{50}$). После этого вы можете выполнять три типа запросов (каждый запрос должен быть выведен на отдельной строке):

- «? x y », где x и y — целые числа ($0 \leq x, y < 2^{50}$). Запрос этого типа служит для использования волшебной коробки. Ответ на него — строка, содержащая число $x \circ y$ ($0 \leq x \circ y < 2^{50}$). Вы можете использовать запрос такого типа не более $4(n + q + q \cdot k)$ раз. **Если вы превысите этот лимит, вам будет выдано число «-1».**
- «next». Запрос такого типа позволяет узнать k элементов массива, которые не любит очередная племянница. Ответ на него — строка из k различных целых чисел d_1, \dots, d_k ($1 \leq d_i \leq n$): индексы элементов, которые не любит эта племянница. *Вы можете снова использовать запрос такого типа только после того, как ответите на предыдущий запрос этого типа.*
- «! x », где x — это целое число ($0 \leq x < 2^{50}$). Здесь x должно быть тем числом числом, которое Джон подарит последней описанной племяннице (то есть племяннице, соответствующей самому недавнему запросу “next”). На этот запрос интерактор не отвечает.

Чтобы предотвратить буферизацию, очищайте буфер после вывода каждой строки: это может быть сделано, например, с помощью `fflush (stdout)` в C или C++, `System.out.flush ()` в Java, `flush (output)` в Pascal или `sys.stdout.flush ()` в Python. А ещё не забывайте завершать каждую строку символом перевода строки.

Пример

стандартный ввод	стандартный вывод
3 3 1	
0 1 0	
1	next
1	? 1 0
2	! 1 next
0	? 0 0
3	! 0 next ! 1

Замечание

В каждом тесте устройство волшебной коробки фиксировано и не зависит от ваших запросов. В разных тестах используются разные устройства. Гарантируется, что волшебная коробка удовлетворяет свойствам, описанным в условии.

В примере операция, которую проделывает волшебная коробка — это операция побитового ИЛИ.