

## Задача A. 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 В. 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 С. 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 помочь 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 подготовил 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 D. Найти количество

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

Дано подвешенное дерево из  $n$  вершин с корнем в 1. Необходимо обработать  $q$  запросов  $(v_i, h_i)$ : найти количество вершин, лежащих в поддереве  $v_i$ , расстояние до которых от  $v_i$  равно  $h_i$ . Расстояние между вершинами — минимальное количество ребер в пути между ними.

### Формат входных данных

Первая строка содержит  $n$  ( $1 \leq n \leq 3 \cdot 10^5$ ).

Следующая строка содержит  $n - 1$  число  $p_2, \dots, p_n$  ( $1 \leq p_i \leq i$ ).  $p_i$  — отец вершины  $i$ .

Следующая строка содержит число  $q$  ( $1 \leq q \leq 3 \cdot 10^5$ ).

Следующие  $q$  строк содержат числа  $v_i, h_i$  ( $1 \leq v_i \leq n, 1 \leq h_i \leq n$ ).

### Формат выходных данных

Для каждого запроса выведите ответ.

### Пример

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

## Задача Е. Вентиляция Лувра

Имя входного файла:	<code>treefirstedge.in</code>
Имя выходного файла:	<code>treefirstedge.out</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Жак — профессиональный вор, который проник в Лувр через систему вентиляции. Однако он заблудился в лабиринте вентиляционных шахт и уже четвёртую неделю пытается найти выход с украденными сокровищами.

Система вентиляции Лувра состоит из  $n$  узлов, соединённых  $n - 1$  вентиляционными шахтами таким образом, что между любыми двумя узлами существует ровно один путь.

Иногда Жак оказывается перед выбором: в какую шахту ползти, чтобы добраться до нужного узла вентиляции. Жак — всего лишь вор, а не инженер, поэтому он не может запомнить все узлы и шахты между ними. Помогите ему узнать, куда ползти.

### Формат входных данных

В первой строке входного файла задано число  $n$  — количество узлов в системе вентиляции Лувра ( $2 \leq n \leq 200\,000$ ).

В следующих  $n - 1$  строках описаны вентиляционные шахты — по одной в строке. Каждая шахта задаётся номерами узлов, которые она соединяет:  $a_i$  и  $b_i$  ( $1 \leq a_i, b_i \leq n$ ;  $a_i \neq b_i$ ). Гарантируется, что между любыми двумя узлами существует единственный путь по шахтам.

В следующей строке задано число  $m$  — количество вопросов Жака ( $1 \leq m \leq 100\,000$ ).

В следующих  $m$  строках описаны вопросы — по одному в строке. Каждый вопрос задаётся номером узла, в котором находится Жак ( $s_i$ ), и номером узла, куда он хочет добраться ( $t_i$ ) ( $1 \leq s_i, t_i \leq n$ ;  $s_i \neq t_i$ ).

Узлы нумеруются с 1.

### Формат выходных данных

Для каждого вопроса выведите номер узла, в который нужно ползти из  $s_i$  напрямую, чтобы попасть в  $t_i$ . Обратите внимание, что ответ единственный, так как между любыми двумя узлами существует ровно один путь.

### Пример

<code>treefirstedge.in</code>	<code>treefirstedge.out</code>
5	
1 2	3
1 3	4
1 4	1
3 5	
3	
5 2	
1 4	
4 3	

## Задача F. Фабиан

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

Маленький Фабиан любит путешествовать и пить кофе. Он хочет выпить кофе в каждом из  $N$  городов своей страны, пронумерованных от 1 до  $N$ . Города соединены ( $N - 1$ ) двусторонними дорогами таким образом, что из любого города можно добраться до любого другого города, двигаясь по этим дорогам.

Фабиан решил пить кофе в каждом городе по порядку номеров: сначала в городе 1, затем в городе 2, и так далее до города  $N$ . Он начинает свой путь из города 1 (где выпивает первую чашку кофе) и отправляется в город 2 за следующей чашкой. Во время этого путешествия он может проходить через несколько других городов, но не будет останавливаться в них, чтобы выпить кофе. После того как он выпьет кофе в городе 2, он отправится в город 3, и так далее, пока наконец не достигнет города  $N$ , где выпьет свою последнюю чашку кофе.

Чтобы пройти по определённой дороге, Фабиану нужен действительный билет. Для  $i$ -й дороги можно купить либо разовый билет стоимостью  $C_{i1}$  евро, либо многоразовый билет стоимостью  $C_{i2}$  евро. Для каждой дороги Фабиан может решить: покупать разовый билет каждый раз, когда ему нужно пройти по этой дороге, или купить многоразовый билет один раз.

Требуется написать программу, которая вычисляет минимальную сумму в евро, которую Фабиану нужно потратить на билеты, чтобы успешно завершить своё путешествие.

### Формат входных данных

В первой строке задано целое число  $N$  ( $2 \leq N \leq 200\,000$ ) — количество городов.

В каждой из следующих ( $N - 1$ ) строк заданы четыре целых числа  $A_i, B_i, C_{i1}, C_{i2}$  ( $1 \leq A_i, B_i \leq N$ ,  $1 \leq C_{i1} \leq C_{i2} \leq 100\,000$ ), которые означают, что города  $A_i$  и  $B_i$  соединены дорогой со стоимостью разового билета  $C_{i1}$  и стоимостью многоразового билета  $C_{i2}$ .

### Формат выходных данных

Выполните одно целое число — минимальную стоимость путешествия.

### Примеры

стандартный ввод	стандартный вывод
5 1 2 2 3 1 3 2 3 1 4 2 3 1 5 2 3	11
4 1 2 3 5 1 3 2 4 2 4 1 3	10
4 1 4 5 5 3 4 4 7 2 4 2 6	16

## Задача G. 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. Диаметром 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	

## Задача Н. Природа преобразилась

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

Коронавирус успел внести свои корректизы буквально во все сферы жизни. Экология — не исключение: стоило крупным промышленным заводам прекратить деятельность, а людям уйти на карантин, как природа вокруг начала преображаться. Во многие города начали возвращаться редкие птицы и животные, а общая экологическая ситуация на планете начала улучшаться.

Вот и у Глеба во дворе выросло дерево и не просто дерево, а дерево с хог-плодами. Как и любое другое дерево, хог-дерево содержит  $n$  ветвей и  $n - 1$  ребро, на каждой ветви растет плод, изначально каждый плод имеет вкус 0. В ходе наблюдений за деревом Глеб заметил, что произошло  $m$  событий двух видов :

1) Глебу стало интересно и он захотел узнать, а какой суммарный вкус имеют все плоды на пути в дереве между плодами  $a$  и  $b$ , к несчастью Глеба интересует не просто сумма, а  $\oplus$ -сумма,  $\oplus$  - операция битового хог

2) У всех плодов на дереве вкус изменился на  $x$ , то есть если до события вкус  $i$ -го плода был  $a_i$ , то теперь он становится  $a_i \oplus x$

### Формат входных данных

В первой строке вводятся два числа :  $n$  — количество вершин в дереве и  $m$  — количество событий, которые произошли с деревом. ( $1 \leq n, m \leq 10^6$ )

На второй строке —  $n - 1$  число, где  $i$ -е число — индекс  $p$  — предка вершины  $i + 1$  ( $1 \leq p \leq i - 1$ ,  $n, m \leq 10^6$ )

Затем следуют  $m$  событий, сначала дается  $t$  - тип запроса, а затем либо  $a, b$  - если требуется найти ответ на запрос первого типа, либо  $x$ , если запрос второго типа. ( $x \leq 10^9, 1 \leq a, b \leq n$ )

### Формат выходных данных

Требуется вывести ответ для каждого запроса первого типа.

### Пример

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

## Задача J. Зомби

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

Зомби-гигант (Giant Zombie) решил, что обычные ночи слишком скучны, и начал полномасштабное вторжение в мир Minecraft. В одиночку бродить по кубическим просторам ему не хотелось, поэтому он призвал на помощь армию своих миньонов.

Зомби-гигант хочет подчинить себе все деревни в округе. Для простоты будем считать, что мир состоит из отдельных деревень. Некоторые деревни соединены двунаправленными тропами. Всего в мире  $n$  деревень и  $n - 1$  тропа. Между любыми двумя деревнями существует единственный простой путь. Иными словами, карта деревень представляет собой дерево.

В ходе вторжения Зомби-гигант заражает некоторые деревни, а некоторые — оставляет в покое (снимает осаду). После каждого действия Зомби-гиганта под влиянием тёмной магии оказываются деревни, принадлежащие минимальному связному подграфу, который содержит все зараженные на данный момент деревни. Обратите внимание, что каждая лично зараженная Гигантом деревня находится под влиянием магии, но не каждая деревня под влиянием магии обязательно была заражена лично Гигантом (она могла просто оказаться на пути между очагами заражения).

В некоторые моменты Стив совершает перебежки из одной деревни в другую. Путешествие считается тем опаснее, чем больше на пути встречается деревень, находящихся под влиянием тёмной магии.

От вас требуется для каждой перебежки Стива определить, сколько находящихся под влиянием магии деревень встретится на пути его следования.

Изначально все деревни свободны от магии.

Если одна или обе конечные деревни маршрута находятся под влиянием магии, то они также учитываются при подсчете.

### Формат входных данных

В первой строке находится одно натуральное число  $n$  ( $1 \leq n \leq 10^5$ ) — количество деревень.

В каждой из следующих  $n - 1$  строке находятся два натуральных числа  $u, v$  ( $1 \leq u, v \leq n, u \neq v$ ) — номера деревень, соединенных тропами. Гарантируется, что заданный граф является деревом.

В следующей строке находится натуральное число  $m$  ( $1 \leq m \leq 10^5$ ) — количество событий. В каждой из следующих  $m$  строк заданы события в хронологическом порядке, в следующем формате: сначала идет натуральное число  $t$  ( $1 \leq t \leq 2$ ) — тип запроса,

- если  $t = 1$ , то далее следует одно натуральное число  $x$  ( $1 \leq x \leq n$ ) — номер деревни, с которой производит действие Зомби-гигант: заражает, если до этого деревня была свободна, и снимает осаду, если до этого деревня была заражена;
- если  $t = 2$ , то далее следуют два натуральных числа  $x, y$  ( $1 \leq x, y \leq n$ ) — номера деревень, между которыми совершает перебежку Стив.

### Формат выходных данных

Для каждой перебежки выведите, сколько деревень под влиянием тёмной магии встретится на пути следования Стива.

## Примеры

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

## Задача К. Стойка

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

Несколько лет назад Министерство инфраструктуры Бергена подготовило план новой сети легкого метро. Эта сеть должна была соединить все  $n$  районов города  $n - 1$  железнодорожными путями таким образом, чтобы существовал путь из любого района в любой другой район. Планируемые пути пронумерованы числами от 1 до  $n - 1$ .

Прошли годы, приближаются новые выборы, а железнодорожная сеть всё ещё существует только на бумаге. Поэтому Министр инфраструктуры (представляющий партию, высоко ценящую разногласия) решил построить хотя бы часть плана. Он попросил каждого из своих  $m$  заместителей выбрать, какие районы, по их мнению, следует соединить. В результате для каждого заместителя министра получился список необходимых путей. Если заместитель министра считает, что районы  $a_1, \dots, a_s$  необходимо соединить, то, по его мнению, необходимыми путями являются все те пути, которые лежат на запланированных маршрутах от  $a_i$  до  $a_j$  для некоторых  $1 \leq i < j \leq s$ .

Министр только что получил все списки от заместителей министров. Он решил в первую очередь построить пути, которые запрашиваются по крайней мере  $k$  заместителей министров. Ваша задача — подготовить список этих путей.

### Формат входных данных

В первой строке входных данных заданы три целых числа  $n$ ,  $m$  и  $k$  ( $2 \leq n \leq 100\,000$ ,  $1 \leq k \leq m \leq 50\,000$ ) — количество районов, количество заместителей министров и минимальное количество запросов соответственно.

В следующих  $n - 1$  строках содержится план; в  $i$ -й из этих строк заданы два целых числа  $a_i$  и  $b_i$  ( $1 \leq a_i, b_i \leq n$ ,  $a_i \neq b_i$ ), означающие, что  $i$ -й путь в плане проходит между районами  $a_i$  и  $b_i$ .

В следующих  $m$  строках заданы районы, выбранные заместителями министров;  $i$ -я из этих строк начинается с целого числа  $s_i$  ( $2 \leq s_i \leq n$ ), которое задаёт количество районов, выбранных  $i$ -м заместителем министра. После него следуют  $s_i$  целых чисел, задающих эти районы.

Гарантируется, что суммарная длина всех списков заместителей министров не превосходит  $100\,000$ , то есть  $\sum_{i=1}^m s_i \leq 100\,000$ .

### Формат выходных данных

В первой строке выходных данных выведите одно целое число  $r$  — количество путей, которые запрашиваются по крайней мере  $k$  заместителей министров.

Во второй строке выведите  $r$  номеров этих путей в возрастающем порядке.

### Пример

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

## Задача L. 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 М. 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

Выполните 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.