

Задача А. Треугольники

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2.5 секунд
Ограничение по памяти: 256 мегабайт

Дан граф. Найдите количество циклов длины 3.

Формат входных данных

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

Каждая из следующих m строк содержит по два целых числа от 1 до n — вершины, которые соединяет соответствующее ребро.

Гарантируется, что в графе нет петель и кратных рёбер.

Формат выходных данных

Выведите одно число — ответ.

Пример

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

Задача В. Частота строки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 512 мегабайт

Дана строка s . Требуется ответить на n запросов. i -й запрос состоит из целого числа k_i и строки m_i , ответом является минимальная длина строки t такой, что t является подстрокой s и строка m_i входит в t как подстрока не менее k_i раз.

Подстрокой строки называется любая последовательность подряд идущих символов в этой строке.

Гарантируется, что для любых двух запросов строки m_i из этих запросов различны.

Формат входных данных

В первой строке содержится строка s ($1 \leq |s| \leq 10^5$).

Во второй строке содержится целое число n ($1 \leq n \leq 10^5$).

В каждой из следующих n строк содержатся целое число k_i ($1 \leq k_i \leq |s|$) и непустая строка m_i — параметры запроса с номером i .

Все строки во вводе состоят только из строчных букв латинского алфавита. Суммарная длина всех строк во вводе не превосходит 10^5 . Все m_i различны.

Формат выходных данных

Для каждого запроса выведите ответ на него в отдельной строке.

Если строка m_i встречается в s менее k_i раз, выведите -1 .

Примеры

стандартный ввод	стандартный вывод
aaaaa	3
5	4
3 a	4
3 aa	-1
2 aaa	5
3 aaaa	
1 aaaaa	
abbb	-1
7	2
4 b	-1
1 ab	3
3 bb	-1
1 abb	1
2 bbb	-1
1 a	
2 abbb	

Задача С. Лунки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.25 секунд
Ограничение по памяти:	256 мегабайт

Маленький Петя очень любит играть. Больше всего он любит играть в игру «Лунки». Это игра для одного игрока со следующими правилами:

Есть N лунок, расположенных в ряд, пронумерованных слева направо числами от 1 до N . У каждой лунки изначально установлена своя сила выброса (у лунки с номером i она равна a_i). Если вбросить шарик в лунку i , то он тут же вылетит из нее и попадет в лунку $i + a_i$, после чего он опять вылетит и т.д.. Если же лунки с таким номером нету, то он просто вылетит за край ряда. На каждом из M ходов игрок выбирает одно из двух действий:

- Установить силу выброса лунки a равной b .
- Вбросить шарик в лунку a и посчитать количество прыжков шарика, прежде чем он вылетит за край ряда, а так же записать номер лунки, после выпрыгивания из которой шарик вылетел за край.

У Пети есть некоторые проблемы с математикой, поэтому, как Вы уже догадались, именно Вам предстоит произвести все подсчеты.

Формат входных данных

Первая строка содержит два числа N и M ($1 \leq N \leq 10^5$, $1 \leq M \leq 10^5$) — количество лунок в ряду и количество ходов. Следующая строка содержит N целых положительных чисел, не превышающих N — начальные силы выброса лунок. Следующие M строк задают ходы, сделанные Петей. Каждая строка может быть двух типов:

- 0 a b
- 1 a

Тут, первый тип означает что требуется установить силу выброса лунки a равной b , а второй означает что требуется вбросить мячик в лунку с номером a . Числа a и b — целые положительные и не превышают N .

Формат выходных данных

Для каждого хода второго типа (задающего вбрасывание шарика) в порядке следования во входном файле выведите два числа через пробел в отдельной строке — номер последней лунки перед вылетом шарика за край и количество прыжков.

Пример

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

Задача D. Запросы композиции перестановок

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

Вам дан массив a_1, \dots, a_n , состоящий из перестановок длины m .

Мы можем определить операцию $+$ для двух перестановок x и y длины m как такую перестановку $z = x + y$, что $z_i = y_{x_i}$ для всех $1 \leq i \leq m$. Заметьте, что порядок сложения важен.

Вам даны q запросов, каждый запрос задается двумя числами $1 \leq l \leq r \leq n$. Рассмотрим перестановку b длины m , такую что $b = ((\dots((a_l + a_{l+1}) + a_{l+2}) + \dots) + a_r)$. Тогда ответом на запрос будет являться сумма $\sum_{i=1}^m i \cdot b_i$. Реализуйте программу, быстро отвечающую на эти запросы.

Формат входных данных

Первая строка содержит единственное целое число t равное количеству тестовых случаев ($1 \leq t \leq 1000$). Далее следует описание t тестовых случаев, каждое в следующем формате:

Первая строка каждого описания содержит два целых числа n, m ($1 \leq n, m \leq 10^5$ и $1 \leq n \cdot m \leq 2 \cdot 10^5$). Следующие n строк содержат по m различных целых чисел $a_{i1}, a_{i2}, \dots, a_{im}$, разделенных пробелами ($1 \leq a_{ij} \leq m$). Следующая строка содержит единственное целое число q ($1 \leq q \leq 2 \cdot 10^5$). Следующие q строк содержат по два целых числа l, r , разделенных пробелами ($1 \leq l \leq r \leq n$).

Гарантируется, что сумма $n \cdot m$ и сумма q по всем тестовым случаям не превосходит $2 \cdot 10^5$.

Формат выходных данных

Выведите ответы на запросы в том порядке, к которому они заданы во входных данных.

Пример

стандартный ввод	стандартный вывод
1	10
4 3	11
3 2 1	11
1 3 2	14
1 2 3	11
2 3 1	
5	
1 1	
1 4	
3 4	
3 3	
1 3	

Задача Е. Машинное обучение

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

На курсе машинного обучения вам выдали первое домашнее задание — вам предстоит проанализировать некоторый массив из n чисел.

В частности, вы интересуетесь так называемой *равномерностью* массива. Предположим, что в массиве число b_1 встречается k_1 раз, b_2 — k_2 раз, и т.д. Тогда *равномерностью* массива называется такое минимальное целое число $c \geq 1$, что $c \neq k_i$ для любого i .

В рамках вашего исследования вы хотите последовательно проделать q операций.

- Операция $t_i = 1, l_i, r_i$ задаёт запрос исследования. Необходимо вывести равномерность массива, состоящего из элементов на позициях от l_i до r_i включительно.
- Операция $t_i = 2, p_i, x_i$ задаёт запрос уточнения данных. Начиная с этого момента времени p_i -му элементу массива присваивается значения x_i .

Формат входных данных

Первая строка содержит n и q ($1 \leq n, q \leq 100\,000$) — размер массива и число запросов соответственно.

Во второй строке записаны ровно n чисел — a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

Каждая из оставшихся q строк задаёт очередной запрос.

Запрос первого типа задаётся тремя числами $t_i = 1, l_i, r_i$, где $1 \leq l_i \leq r_i \leq n$ — границы соответствующего отрезка.

Запрос второго типа задаётся тремя числами $t_i = 2, p_i, x_i$, где $1 \leq p_i \leq n$ — позиция в которой нужно заменить число, а $1 \leq x_i \leq 10^9$ — его новое значение

Формат выходных данных

Для каждого запроса первого типа выведите одно число — равномерность соответствующего отрезка массива.

Пример

стандартный ввод	стандартный вывод
10 4	2
1 2 3 1 1 2 2 2 9 9	3
1 1 1	2
1 2 8	
2 7 1	
1 2 8	

Замечание

Первый запрос состоит из ровно одного элемента — 1. Минимальное подходящее $c = 2$.

Отрезок второго запроса состоит из четырёх 2, одной 3 и двух 1. Минимальное подходящее $c = 3$.

Отрезок четвёртого запроса состоит из трёх 1, трёх 2 и одной 3. Минимальное подходящее $c = 2$.

Задача F. Общий сосед

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

Изначально есть граф на N вершинах без рёбер. Обработайте Q запросов.

- 1 u v — добавить в граф ребро, соединяющее вершины u и v ;
- 2 u v — определить номер вершины, которая соединена ребром и с u , и с v , или сообщить, что такой нет.

Гарантируется, что в графе **никогда** не появятся циклы.

Запросы даны в зашифрованном виде. А именно, вместо явно заданных параметров запроса t, u, v (где $t \in \{1, 2\}$ — тип), Вам будут даны параметры a, b, c . Пусть X это ответ на предыдущий запрос ($X = 0$ для первого запроса). Тогда реальные параметры запроса могут быть вычислены по формулам

- $t = 1 + ((a \cdot (1 + X)) \bmod 998\,244\,353) \bmod 2$;
- $u = 1 + ((b \cdot (1 + X)) \bmod 998\,244\,353) \bmod N$;
- $v = 1 + ((c \cdot (1 + X)) \bmod 998\,244\,353) \bmod N$;

Формат входных данных

В первой строке входных данных вводятся два числа N и Q — количество вершин и запросов ($2 \leq N \leq 10^5, 1 \leq Q \leq 10^5$).

В каждой из следующих строк вводятся зашифрованные параметры запросов a, b, c ($0 \leq a, b, c < 998\,244\,353$). Гарантируется, что в реальных запросах $u \neq v$, а также что в графе никогда не появятся циклы.

Формат выходных данных

На каждый запрос второго типа выведите одно число — 0, если нет общего соседа у u и v , и номер общего соседа, иначе. Можно показать, что при заданных ограничениях номер общего соседа определяется однозначно.

Примеры

стандартный ввод	стандартный вывод
6 12	0
143209915 123840720 97293110	2
89822758 207184717 689046893	0
67757230 270920641 26993265	2
952858464 221010240 871605818	6
730183361 147726243 445163345	0
963432357 295317852 195433903	1
120904472 106195318 615645575	
817920568 27584394 770578609	
38727673 250957656 506822697	
139174867 566158852 412971999	
205467538 606353836 855642999	
159292205 319166257 51234344	
2 1	
377373366 41280240 33617925	

Замечание

В примере расшифрованные запросы это:

6 12
2 1 3
1 2 6
1 2 4
1 1 3
2 4 6
2 1 4
1 5 6
1 1 2
2 1 4
2 2 5
2 3 4
2 2 3

Задача G. Покраска соседей

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мегабайта

Дан неориентированный граф на N вершинах и M рёбрах. Вершины пронумерованы от 1 до N , и изначально вершина с номером i покрашена в цвет i . Над графом выполняются Q операций, каждая из которых описывается одной вершиной v_i . Во время такой операции происходят следующие действия:

- Пусть C это текущий цвет вершины v ;
- Все вершины u , соединенные с v ребром, перекрашиваются в цвет C .

Определите цвета всех вершин после всех операций.

Формат входных данных

В первой строке входных данных даны три целых числа N, M, Q — количество вершин в графе, количество рёбер и число производимых над графом операций, соответственно ($1 \leq N, Q \leq 2 \cdot 10^5$, $0 \leq M \leq \min(2 \cdot 10^5, \frac{N(N-1)}{2})$).

В каждой из следующих M строк даны номера двух вершин u и v — концы очередного ребра ($1 \leq u, v \leq N$, $u \neq v$).

В последней строке даны Q чисел v_1, \dots, v_Q — номера вершин, с которыми производятся операции ($1 \leq v_i \leq N$).

Формат выходных данных

В единственной строке выведите N целых чисел, i -е из них должно быть цветом вершины i после выполнения всех операций.

Примеры

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

Задача Н. Компоненты связности на подотрезках

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3.5 секунд
Ограничение по памяти:	256 мегабайт

Для графа на N вершинах дан список рёбер: $(u_1, v_1), \dots, (u_M, v_M)$. От Вас требуется ответить на Q запросов вида $[L_j, R_j]$. Ответ на j -й запрос это количество компонент связности, которое будет в графе, если оставить в нём только рёбра с номерами от L_j до R_j .

Формат входных данных

В первой строке входных данных даны два числа N и M ($1 \leq N, M \leq 10^5$).

В i -й из следующих M строк даны номера двух вершин u_i, v_i , соединённых i -м ребром ($1 \leq u_i, v_i \leq N$).

В следующей строке вводится число Q — количество запросов ($1 \leq Q \leq 10^5$).

В j -й из следующих Q строк вводятся два числа L_j, R_j — границы отрезка для j -го запроса ($1 \leq L_j \leq R_j \leq M$).

Формат выходных данных

Для j -го запроса выведите одно число — количество компонент связности в графе на рёбрах с номерами из подотрезка $[L_j, R_j]$.

Примеры

стандартный ввод	стандартный вывод
4 4 2 3 1 2 3 4 4 1 4 2 2 2 3 1 4 2 4	3 2 1 1
10 10 9 1 3 10 8 4 5 8 3 9 3 9 3 10 1 9 3 5 10 7 10 1 5 5 8 3 8 3 5 5 7 3 4 2 8 5 9 5 8 7 8	5 7 5 7 8 8 5 6 7 8

Задача I. Польшар и Подарки

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

Рождество! Польшар и его друзья будут дарить друг другу подарки. Всего шаров n . Каждый шар должен подарить подарок ровно одному другому шару в соответствии с некоторой перестановкой p , $p_i \neq i$ для всех i .

К сожалению, шары забывчивы. Мы знаем, что ровно k шаров забудут принести свои подарки. Шар номер i получит подарок, если будут выполнены следующие два условия:

1. Шар номер i должен принести свой подарок.
2. Шар x такой, что $p_x = i$, должен принести свой подарок.

Какое минимально и максимально возможное число шаров, которые **не** получают свой подарок, если ровно k шаров забудут принести свой подарок?

Формат входных данных

В первой строке находится два целых числа n и k ($2 \leq n \leq 10^6$, $0 \leq k \leq n$) — общее число шаров и число шаров, которые забудут подарки.

Во второй строке находится перестановка p целых чисел от 1 до n , где p_i — номер шара, которому должен дать подарок шар номер i . Для всех i выполняется $p_i \neq i$.

Формат выходных данных

Выведите два числа — минимально и максимально возможное число шаров, которые **не** получают подарков, соответственно.

Примеры

стандартный ввод	стандартный вывод
5 2 3 4 1 5 2	2 4
10 1 2 3 4 5 6 7 8 9 10 1	2 2

Замечание

В первом примере, если первый и третий шары забудут принести подарок, то они же и будут единственными, кто не получит подарка. Поэтому минимальный ответ равен 2. Однако, если первый и второй шары забудут, то только пятый шар получит подарок. Поэтому максимальный ответ равен 4.

Задача J. Танцующие слоны

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

Танцующие слоны — это зрелищное шоу в Паттайе, в котором участвуют N слонов, танцующих на одной линии, называемой сценой.

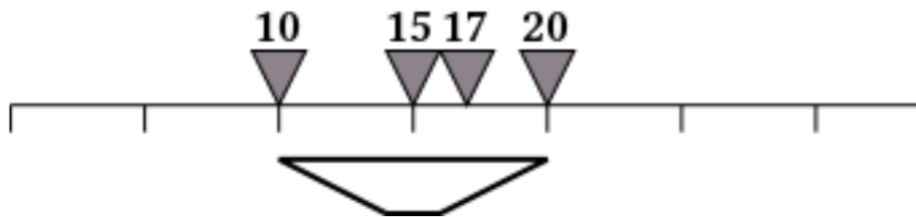
В результате многолетних тренировок слоны, участвующие в шоу, разучили большое количество танцевальных движений. Все шоу состоит из последовательности актов. В каждом акте только один слон совершает одно танцевальное движение, в результате которого он может переместиться на другую позицию на сцене.

Постановщики шоу хотят сделать фотоальбом, который бы содержал фотографии всего шоу. После каждого акта они хотят сделать фотографии всех слонов.

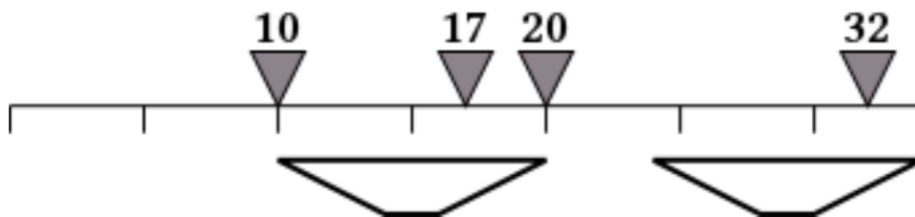
В любой момент времени на протяжении шоу некоторое количество слонов может находиться в одной и той же позиции — это значит, что слоны просто стоят рядом.

Одна фотокамера может фотографировать группу слонов тогда и только тогда, когда все позиции, в которых находятся слоны, лежат на отрезке длины L (обе границы отрезка включаются в него). Так как слоны могут располагаться вдоль всей сцены, то может потребоваться несколько фотокамер, чтобы сфотографировать всех слонов одновременно.

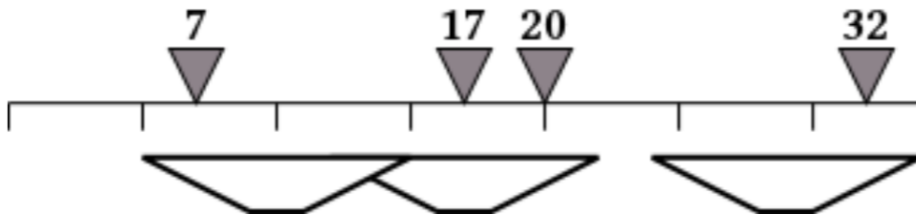
К примеру, предположим, что $L = 10$ и слоны располагаются на сцене в позициях 10, 15, 17, и 20 соответственно. В этот момент достаточно одной фотокамеры, чтобы сфотографировать всех слонов, как это показано ниже. (Слоны изображены как треугольники; фотокамеры изображены как трапеции).



В последующем акте слон, находящийся в позиции 15, в результате танцевального движения перемещается в позицию 32. После этого акта необходимо уже не менее двух фотокамер для того, чтобы сфотографи



В следующем акте слон, находящийся в позиции 10, перемещается в позицию 7. В данном случае понадобится 3 фо:



В данной задаче вы должны определить минимальное количество фотокамер, необходимых для того, чтобы сделать фотографии после каждого акта шоу. Следует отметить, что количество необходимых фотокамер может увеличиваться, уменьшаться, или оставаться тем же самым от акта к акту.

Формат входных данных

В первой строке даны три целых числа N , M , и L ($1 \leq N, M \leq 150\,000$, $1 \leq L \leq 10^9$) — количество слонов, количество актов и длина отрезка, охватываемого одной фотокамерой.

Следующая строка содержит N целых неотрицательных чисел, не превосходящих 10^9 — позиции слонов, i -е из них задаёт позицию слона $i - 1$ (слоны нумеруются с 0).

В каждой из следующих M строк задаётся информация об актах. В i -й строке даны 2 числа A_i и B_i ($1 \leq A_i, B_i \leq 10^9$). Из этих чисел i -й запрос генерируется следующим образом: номер слона, который перемещается $X_i = (A_i + C) \% N$, где C — ответ на предыдущий запрос если $i > 1$, и $C = 0$ если $i = 1$, а $\%$ означает взятие по модулю. Позиция, на которую перемещается слон с номером X_i в i -м акте равна B_i .

Формат выходных данных

В единственной строке выведите M чисел, i -е из них — минимальное количество фотокамер, необходимых для фотографирования после i -го акта.

Пример

стандартный ввод	стандартный вывод
4 5 10 10 15 17 20 2 16 0 25 1 35 2 38 0 0	1 2 2 2 3

Задача К. Простые хог-разбиения

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	10 секунд
Ограничение по памяти:	512 мегабайт

Вам дан массив натуральных чисел. Ваша задача заключается в том, чтобы обрабатывать два типа запросов:

- Изменить значение элемента массива
- По заданному небольшому множеству чисел сказать, можно ли разбить весь массив на какое-то количество непересекающихся подотрезков, что на каждом отрезке «исключающее или» (XOR) значений принадлежит этому множеству. Каждый элемент массива должен подлежать какому-то отрезку.

Заметьте, что не обязательно, чтобы каждое число из заданного множества было равно «исключающему или» какого-либо из отрезков, а также любое число из множества может быть равно «исключающему или» любого количества подотрезков. Например, если «исключающее или» всех чисел в массиве принадлежит множеству, можно выбрать весь массив в качестве единственного подотрезка.

Формат входных данных

Первая строка содержит два числа n и q ($1 \leq n \leq 10^5$; $1 \leq q$) — длина массива и количество запросов.

Вторая строка содержит n чисел x_i ($0 \leq x_i < 2^{20}$) — изначальный массив. Следующие q строк описывают запросы. Каждый запрос может быть одного из двух типов.

Запросы первого типа задаются в формате $1 \ a \ b$ ($1 \leq a \leq n$; $0 \leq b < 2^{20}$)? где a это индекс изменяемого элемента, а b это его новое значение.

Запросы второго типа задаются в формате $2 \ k \ a_1 \ \dots \ a_k$ ($1 \leq k \leq 5$; $0 \leq a_i < 2^{20}$) где k это размер множества, а a_i это его элементы. Заметьте, что числа a_i не обязательно различны.

Количество запросов первого типа не превосходит $4 \cdot 10^5$, и сумма значений k по всем запросам второго типа не превосходит 10^5 .

Формат выходных данных

Для каждого запроса второго типа выведите одну строку в таком же порядке, как они следуют во входе. Если разбить массив в соответствующем запросе возможно, выведите **ТАК** («да» по-польски), иначе выведите **НІЕ** («нет» по-польски).

Пример

стандартный ввод	стандартный вывод
5 10	ТАК
1 2 0 3 0	ТАК
2 1 3	ТАК
2 1 0	НІЕ
1 3 5	
2 2 6 3	
1 1 8	
1 2 5	
1 3 3	
1 4 1	
1 5 1	
2 3 2 4 8	

Замечание

В первом запросе второго типа массив равен $\{1, 2, 0, 3, 0\}$, поэтому его можно разбить следующим образом: $\{1, 2, 0\}$ и $\{3, 0\}$.

Во втором запросе первого типа массив такой же, и «исключающее или» всех элементов равен нулю, поэтому можно выбрать весь массив в качестве подотрезка.

В третьем запросе второго типа массив равен $\{1, 2, 5, 3, 0\}$, и его можно разбить на $\{1, 2, 5\}$ и $\{3, 0\}$.

В последнем запросе разбить массив невозможно.

Задача L. МЕХ на пути

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

Дано дерево, на каждом ребре которого написано неотрицательное целое число. Вам необходимо ответить на несколько запросов вида «для данных вершин u , v назовите наименьшее неотрицательное целое число, которое **не** встречается среди чисел, написанных на ребрах на пути от u до v ».

Формат входных данных

Первая строка входных данных содержит два числа n и q ($2 \leq n \leq 10^5$, $1 \leq q \leq 10^5$), количество вершин и количество запросов.

Следующие $n - 1$ строк содержат по три числа u_i, v_i, x_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$, $0 \leq x_i \leq 10^9$), которые описывают ребро дерева (u_i, v_i) , на котором написано число x_i .

Следующие q строк содержат по паре чисел a_j, b_j ($1 \leq a_j, b_j \leq n$), которая обозначает запрос на пути от a_j до b_j .

Формат выходных данных

Для каждого запроса выведите единственное число — минимальное число, которое не встречается на пути.

Пример

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

Задача М. Кенгуру

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

Даня хочет поехать в Австралию и заняться там фотографированием кенгуру. На своём пути он n раз остановится у каких-то точек наблюдения. Известно, что у i -й точки наблюдения кенгуру есть на расстоянии от a_i до b_i метров от неё.

Также у Дани есть m объективов для фотоаппарата. Известно, что i -й объектив позволяет делать фотографии объектов, находящихся на расстоянии от l_i до r_i метров.

Таким образом, i -ю линзу можно использовать у j -й точки наблюдения, если отрезки $[l_i; r_i]$ и $[a_j; b_j]$ имеют общую точку.

Для каждой линзы найдите длину самого длинного непрерывного отрезка точек наблюдения, на котором её можно использовать.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n \leq 50000$, $1 \leq m \leq 2 \cdot 10^5$) — количество точек наблюдения и количество линз, соответственно.

Следующие n строк содержат описание точек наблюдения. Каждая из этих строк содержит два целых числа a_i и b_i ($1 \leq a_i \leq b_i \leq 10^9$) — минимальная и максимальная дистанция, на которой будут кенгуру, для i -й точки наблюдения.

Следующие m строк содержат описания линз. Каждая из этих строк содержит два целых числа l_i и r_i ($1 \leq l_i \leq r_i$) — минимальная и максимальная дистанция, на которой можно наблюдать кенгуру, используя i -ю линзу.

Формат выходных данных

Выведите m строк, i -я должна содержать количество точек наблюдения в самой длинном непрерывном отрезке точек наблюдения таком, что на нём Даня может использовать i -ю линзу.

Пример

стандартный ввод	стандартный вывод
3 3	2
2 5	3
1 3	0
6 6	
3 5	
1 10	
7 9	

Задача N. Сериал по Майнкрафту

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

Маленький Миша ходит на кружок по программированию и ничего там не решает. Это может показаться странным, но когда вы узнаете, что Миша снимает сериал по Майнкрафту, все сразу встанет на свои места...

Миша, вдохновляясь застройкой Манхэттена, построил в Майнкрафте город, который можно представить в виде таблицы $n \times m$. В городе живут k школьников, i -й школьник живет в доме, который находится на пересечении x_i -й строки и y_i -го столбца. Также у каждого школьника есть степень его агрессивности w_i . Так как город оказался очень большим, Миша решил территориально ограничить действия своего сериала некоторым принадлежащим таблице квадратом s , стороны которого параллельны осям координат и имеют длину от 1 до $\min(n, m)$ клеток.

По сюжету главный герой приедет в город и сразу же попадет в квадрат s . Обладая уникальной степенью агрессивности 0 он сможет проявить свои лидерские качества и собрать команду из спокойных, умеренных и агрессивных школьников.

Чтобы собранная команда была разносторонней, но сплоченной, агрессивности всех школьников в ней должны быть попарно различны и должны образовывать единый отрезок подряд идущих целых чисел. То есть, если внутри квадрата s **найдутся** школьники со степенями агрессивности $l, l+1, \dots, -1, 1, \dots, r-1, r$, где $l \leq 0 \leq r$, то главный герой сможет собрать команду из $r-l+1$ человека (сам он тоже входит в эту команду).

Обратите внимание, брать в команду всех школьников из квадрата s не обязательно.

Миша считает, что в команде главного героя должно быть хотя бы t человек. Поэтому его интересует, сколько существует квадратов в таблице, попав в которые, главный герой сможет набрать команду как минимум из t человек. Помогите Мише это посчитать.

Формат входных данных

Первая строка содержит четыре целых числа n, m, k и t ($1 \leq n, m \leq 40\,000$, $1 \leq n \cdot m \leq 40\,000$, $1 \leq k \leq 10^6$, $1 \leq t \leq k+1$) — количество строк в таблице, количество столбцов, количество школьников, которые живут в городе и необходимый размер команды соответственно.

Каждая из k следующих строк содержит по три целых числа x_i, y_i и w_i ($1 \leq x_i \leq n$, $1 \leq y_i \leq m$, $1 \leq |w_i| \leq 10^9$) — номер строки и номер столбца, на пересечении которых живет i -й школьник, а так же степень его агрессивности.

Формат выходных данных

Выведите одно целое число — количество способов выбрать квадрат s таким образом, чтобы главный герой смог набрать команду, состоящую, хотя бы из t человек.

Примеры

стандартный ввод	стандартный вывод
2 2 1 2 1 1 2	0
2 2 2 2 1 1 1 2 2 2	2
2 2 4 2 1 1 1 1 1 -1 1 2 1 2 2 1	4

Замечание

1. В первом тестовом примере главный герой ни в каком выбранном квадрате s не сможет набрать команду, состоящую из более чем одного человека.

2	

Иллюстрация к первому тестовому примеру.

2. Во втором тестовом примере можно выбрать квадрат s двумя способами, изображенными ниже, в одном из них главный герой сможет набраться команду из школьников с степенями агрессивности $[0, 1]$, а в другом — $[0, 1, 2]$. Обратите внимание на то, что вне зависимости от выбранного квадрата главный герой со степенью агрессивности 0 всегда будет входить в команду.

1	
	2

1	
	2

Иллюстрация ко второму тестовому примеру.

3. В третьем тестовом примере можно выбрать квадрат s четырьмя способами, изображенными ниже, в них главный герой сможет набраться команды со следующими степенями агрессивности школьников, соответственно: $[-1, 0, 1]$, $[0, 1]$, $[0, 1]$, $[-1, 0, 1]$.

1,-1	1
	1

1,-1	1
	1

1,-1	1
	1

1,-1	1
	1

Иллюстрация к третьему тестовому примеру.