

Задача А. 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 В. Мы в некотором роде не совсем гавайцы

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

Миновав множество опасностей и потеряв студента Лома, экипаж «Беды» прибывает на Гавайи. Предприимчивый директор театра принимает путешественников за артистов и просит выступить в местном театре.

В гримёрке в ряд расположены n чехлов от контрабасов с высотами $a_1, a_2, \dots, a_n$. Несчастный Фукс забыл, какой из чехлов принадлежит ему, поэтому просит вашей помощи.

Назовём *схожестью* двух мультимножеств количество элементов в их пересечении, например, *схожесть* мультимножеств $\{1, 4, 4, 5\}$ и $\{2, 4, 5, 7, 11\}$ равна 2 (пересечение $\{4, 5\}$), а *схожесть* мультимножеств $\{3, 3, 3\}$ и $\{1, 3, 3, 3, 4\}$ равна 3 (пересечение $\{3, 3, 3\}$). Обратите внимание, что порядок элементов в мультимножестве не имеет значения.

Вам предстоит ответить на q запросов. Каждый запрос задаёт подотрезок чехлов $a_l, a_{l+1}, \dots, a_r$. Вы можете выбрать любое целое число k ($l \leq k \leq r$), чтобы *схожесть* подотрезка $a_l, a_{l+1}, \dots, a_k$ и подотрезка $a_{k+1}, \dots, a_r$ была **максимальной**. Найдите максимальную возможную *схожесть*.

Вы не понимаете, как такие странные запросы помогут Фуксу с поиском своего чехла, но соглашаетесь помочь.



Формат входных данных

В первой строке дано целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных.

В первой строке каждого набора дано целое число n ($1 \leq n \leq 10^5$) — количество чехлов от контрабасов.

Во второй строке каждого набора даны n целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$) — высоты чехлов.

В третьей строке каждого набора дано целое число q ($1 \leq q \leq 10^5$) — количество запросов.

В следующих q строках даны целые числа l и r ($1 \leq l \leq r \leq n$) — запросы поиска максимальной *схожести*.

Гарантируется, что сумма n по всем наборам входных данных не превосходит 10^5 . То же самое гарантируется для q .

Формат выходных данных

Для каждого набора входных данных выведите q целых чисел — ответы на соответствующие запросы.

Пример

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

Задача С. Поиск общего соседа

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

К графу на N вершинах, в котором нет рёбер, поступают Q запросов двух типов:

- 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. Таким образом, ответ на второй запрос определяется однозначно. Если общего соседа у вершин u и v не существует, то ответом на второй запрос считается число 0.

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 второго типа ($X = 0$ 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 расшифрованных 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.

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 D. 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 Е. Ограниченность произведений

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

Даны целые числа N и K . Найдите количество последовательностей из K положительных целых чисел, в которых произведение любых двух соседних элементов не превышает N . Ответ выведите по модулю $10^9 + 7$.

Формат входных данных

В единственной строке входных данных вводятся целые числа N и K — ограничение на произведение соседних элементов и длина последовательности ($1 \leq N \leq 10^9$, $2 \leq K \leq 1\,000$).

Формат выходных данных

Выведите одно целое число — количество последовательностей, удовлетворяющих условию, по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
3 2	5
314159265 35	457397712
6 7	5073
1 1000	1

Замечание

В первом примере подходят последовательности $(1, 1)$, $(1, 2)$, $(1, 3)$, $(2, 1)$, $(3, 1)$.

Задача F. На обратной стороне карточки

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

Даны N карточек, пронумерованных от 1 до N . На лицевой стороне карточки i написано целое число A_i , а на обратной — целое число B_i . Сумма всех чисел на обеих сторонах всех карточек равна M .

Изначально все карточки лежат лицевой стороной вверх. Можно перевернуть любое количество карточек. Для каждого целого числа k от 0 до M найдите минимальное количество карточек, которые нужно перевернуть, чтобы сумма чисел на видимых сторонах стала в точности равна k .

Формат входных данных

В первой строке входных данных вводятся целые числа N и M — количество карточек и сумма всех чисел на обеих сторонах всех карточек ($1 \leq N \leq 200\,000$, $0 \leq M \leq 200\,000$).

В i -й из следующих N строк вводятся целые числа A_i и B_i — числа на лицевой и обратной сторонах карточки i ($0 \leq A_i, B_i \leq M$).

Гарантируется, что $\sum_{i=1}^N (A_i + B_i) = M$.

Формат выходных данных

Выведите $M + 1$ строку. В $(k + 1)$ -й строке выведите минимальное количество карточек, которые нужно перевернуть, чтобы сумма чисел на видимых сторонах стала равна k . Если получить такую сумму невозможно, выведите -1 .

Примеры

стандартный ввод	стандартный вывод
3 6 0 2 1 0 0 3	1 0 2 1 1 3 2
2 3 1 1 0 1	-1 0 1 -1
5 12 0 1 0 3 1 0 0 5 0 2	1 0 1 1 1 1 2 1 2 2 2 3 3 4

Замечание

В первом примере для получения суммы 0 достаточно перевернуть вторую карточку. Для полу-

чения суммы 5 нужно перевернуть все три карточки.

Задача G. 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 Н. Офис

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

В офисе одной IT-компании работают N сотрудников, пронумерованных от 1 до N . Поскольку это IT-компания, все входы и выходы сотрудников записываются в журнал. За день в журнале появились M записей. Запись i состоит из двух целых чисел T_i и P_i и означает, что в момент времени T_i сотрудник P_i вошёл в офис, если находился снаружи, или вышел из офиса, если находился внутри.

К счастью, все сотрудники соблюдают баланс работы и личной жизни, так что до первого события и после последнего события в журнале все сотрудники находятся снаружи офиса.

Некоторые пары сотрудников хотят вместе выпить кофе, но для этого им нужно оказаться одновременно в офисе. Помогите им узнать, сколько времени у них было для такой встречи. Даны Q запросов. В запросе i даны номера двух сотрудников A_i и B_i . Определите суммарную продолжительность времени, в течение которого оба сотрудника одновременно находились в офисе и могли вместе выпить кофе.

Формат входных данных

В первой строке входных данных вводятся целые числа N и M — количество сотрудников и записей в журнале ($2 \leq N, M \leq 200\,000$).

В i -й из следующих M строк вводятся целые числа T_i и P_i — время события и номер сотрудника ($1 \leq T_i \leq 10^9, 1 \leq P_i \leq N$).

Гарантируется, что $T_1 < T_2 < \dots < T_M$ и каждый сотрудник встречается в журнале чётное количество раз.

В следующей строке вводится целое число Q — количество запросов ($1 \leq Q \leq 200\,000$).

В i -й из следующих Q строк вводятся целые числа A_i и B_i — номера сотрудников в запросе i ($1 \leq A_i < B_i \leq N$).

Формат выходных данных

Выведите Q строк. В i -й строке выведите одно целое число — суммарную продолжительность времени, в течение которого сотрудники A_i и B_i одновременно находились в офисе и могли вместе выпить кофе.

Примеры

стандартный ввод	стандартный вывод
3 8 10 1 20 2 30 1 40 3 60 3 70 1 80 2 90 1 3 1 2 1 3 2 3	20 0 20
4 12 1 1 2 2 3 3 4 2 5 4 6 1 8 2 10 1 11 3 13 2 14 4 15 1 6 1 2 1 3 1 4 2 3 2 4 3 4	5 4 5 4 5 6

Замечание

В первом примере первый и второй сотрудники одновременно находились в офисе с момента 20 до момента 30 и с момента 70 до момента 80. Суммарная продолжительность равна 20.

Первый и третий сотрудники никогда не находились в офисе одновременно, поэтому ответ на второй запрос равен 0.

Второй и третий сотрудники одновременно находились в офисе с момента 40 до момента 60, поэтому ответ на третий запрос равен 20.

Задача I. 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 J. Множества прыжков

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

Даны N клеток, расположенных в ряд и пронумерованных от 1 до N , и массив A из N положительных целых чисел. Изначально первая клетка покрашена в чёрный цвет, остальные — в белый. На первой клетке стоит фишка.

За одну операцию можно выбрать положительное целое число x и переместить фишку из клетки i в клетку $i + A_i \cdot x$, если её номер не превышает N . После этого клетка, в которую переместилась фишка, красится в чёрный цвет.

Можно выполнить любое количество операций, в том числе ни одной. Найдите количество различных множеств чёрных клеток, которые можно получить, по модулю 998 244 353.

Формат входных данных

В первой строке входных данных вводится целое число N — количество клеток ($1 \leq N \leq 200\,000$).
Во второй строке вводятся N целых чисел $A_1, A_2, \dots, A_N$ ($1 \leq A_i \leq 200\,000$).

Формат выходных данных

Выведите одно целое число — количество различных множеств чёрных клеток, которые можно получить, по модулю 998 244 353.

Примеры

стандартный ввод	стандартный вывод
5 1 2 3 1 1	8
40 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	721419738
10 2 1 3 2 1 4 2 1 3 1	18

Замечание

В первом примере можно получить восемь множеств чёрных клеток: $\{1\}$, $\{1, 2\}$, $\{1, 2, 4\}$, $\{1, 2, 4, 5\}$, $\{1, 3\}$, $\{1, 4\}$, $\{1, 4, 5\}$ и $\{1, 5\}$.

Задача К. 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.

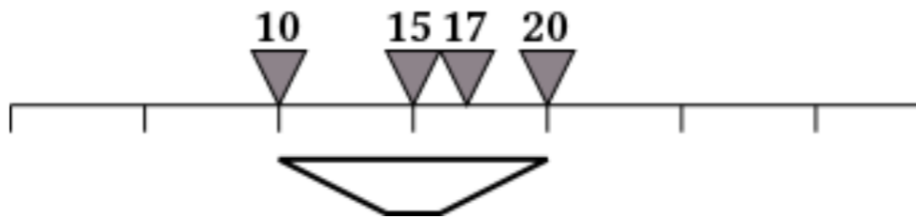
В результате многолетних тренировок слоны, участвующие в шоу, разучили большое количество танцевальных движений. Все шоу состоит из последовательности актов. В каждом акте только один слон совершает одно танцевальное движение, в результате которого он может переместиться на другую позицию на сцене.

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

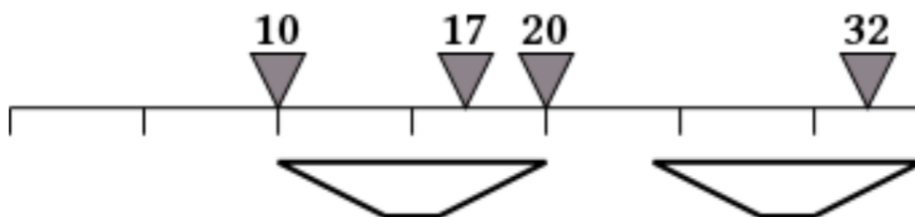
В любой момент времени на протяжении шоу некоторое количество слонов может находиться в одной и той же позиции — это значит, что слоны просто стоят рядом.

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.

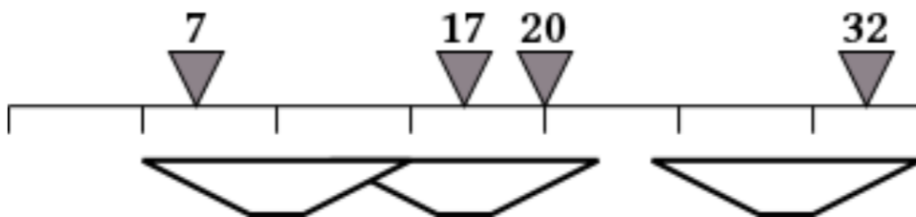
К примеру, предположим, что $L = 10$ и слоны располагаются на сцене в позициях 10, 15, 17, и 20 соответственно. В этот момент достаточно одной фотокамеры, чтобы сфотографировать всех слонов, как это показано ниже. (Слоны изображены как треугольники; фотокамеры изображены как трапеции).



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 L. Запросы на порождённых подграфах

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

Дан неориентированный граф G из n вершин и m рёбер без петель и кратных рёбер. Вершины пронумерованы от 1 до n .

Нужно ответить на q запросов. В запросе даны числа l, r и k . Рассмотрим подграф, состоящий из вершин $l, l+1, \dots, r$ и всех рёбер исходного графа между ними. Для каждой его вершины вычислим побитовое исключающее ИЛИ номеров её соседей. Для вершины без соседей это значение равно 0.

Найдите k -е по возрастанию среди полученных значений. Повторяющиеся значения учитываются отдельно.

Формат входных данных

В первой строке вводится количество наборов входных данных t ($1 \leq t \leq 15\,000$).

В каждом наборе сначала вводятся числа n и m ($2 \leq n \leq 150\,000$, $1 \leq m \leq 150\,000$).

В следующих m строках вводятся концы рёбер u_i и v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$). Гарантируется отсутствие кратных рёбер.

Далее вводится количество запросов q ($1 \leq q \leq 150\,000$).

В следующих q строках вводятся числа l, r и k ($1 \leq l \leq r \leq n$, $1 \leq k \leq r - l + 1$).

Суммы n , m и q по всем наборам не превышают 150 000 каждая.

Формат выходных данных

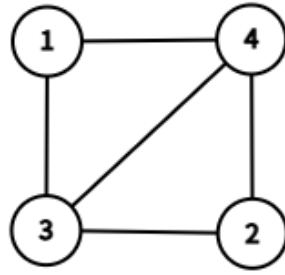
Для каждого набора выведите q целых чисел — ответы на запросы в порядке их поступления.

Пример

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

Замечание

В первом тесте исходный граф изображён на рисунке ниже.

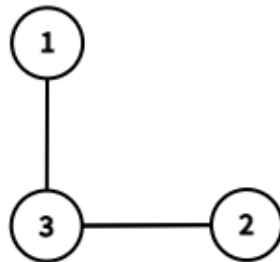


В первом запросе рассматривается подграф на вершинах 1 и 2.



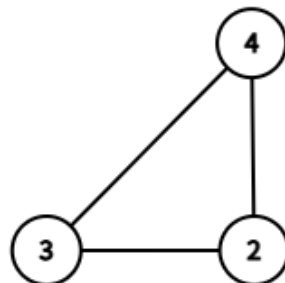
Обе вершины не имеют соседей, поэтому их значения равны 0. Второе по возрастанию значение равно 0.

Во втором запросе рассматривается подграф на вершинах 1, 2 и 3.



Значения вершин 1 и 2 равны 3, а значение вершины 3 равно $1 \oplus 2 = 3$. Первое по возрастанию значение равно 3.

В третьем запросе рассматривается подграф на вершинах 2, 3 и 4.



Значения вершин 2, 3 и 4 равны соответственно $3 \oplus 4 = 7$, $2 \oplus 4 = 6$ и $2 \oplus 3 = 1$. В порядке возрастания получаем 1, 6, 7, поэтому ответ равен 7.

Задача М. Простые xor-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 N. 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	