

Задача А. Разреженные таблицы

Имя входного файла: `sparse.in`
Имя выходного файла: `sparse.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан массив из n чисел. Требуется написать программу, которая будет отвечать на запросы следующего вида: найти минимум на отрезке между u и v включительно.

Формат входных данных

В первой строке входного файла даны три натуральных числа n , m ($1 \leq n \leq 10^5$, $1 \leq m \leq 10^7$) и a_1 ($0 \leq a_1 < 16\,714\,589$) — количество элементов в массиве, количество запросов и первый элемент массива соответственно. Вторая строка содержит два натуральных числа u_1 и v_1 ($1 \leq u_1, v_1 \leq n$) — первый запрос.

Элементы a_2, a_3, \dots, a_n задаются следующей формулой:

$$a_{i+1} = (23 \cdot a_i + 21563) \bmod 16714589.$$

Например, при $n = 10$, $a_1 = 12345$ получается следующий массив: $a = (12345, 305498, 7048017, 11694653, 1565158, 2591019, 9471233, 570265, 13137658, 1325095)$.

Запросы генерируются следующим образом:

$$\begin{aligned} u_{i+1} &= ((17 \cdot u_i + 751 + ans_i + 2i) \bmod n) + 1, \\ v_{i+1} &= ((13 \cdot v_i + 593 + ans_i + 5i) \bmod n) + 1, \end{aligned}$$

где ans_i — ответ на запрос номер i .

Обратите внимание, что u_i может быть больше, чем v_i .

Формат выходных данных

В выходной файл выведите u_m , v_m и ans_m (последний запрос и ответ на него).

Примеры

<code>sparse.in</code>	<code>sparse.out</code>
10 8 12345 3 9	5 3 1565158

Задача В. Друзья и последовательности

Имя входного файла: friends.in
Имя выходного файла: friends.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Майк и !Майк соперничают еще со школьных лет, они противоположны во всем что делают, кроме программирования. Сегодня у них возникла проблема, которую сами друзья сами решить не могут, но вместе с вами — кто знает?

Каждый из них знает две последовательности n чисел a и b . По запросу в виде пары целых чисел (l, r) Майк может сразу сообщить значение $\max_{i=l}^r a_i$, а !Майк — значение $\min_{i=l}^r b_i$.

Предположим, что робот задает им каждый из возможных различных запросов в виде пары целых чисел (l, r) ($1 \leq l \leq r \leq n$) (то есть он сделает ровно $n(n+1)/2$ запросов) и считает, сколько раз их ответы на один и тот же запрос совпадают, то есть для скольких пар выполняется $\max_{i=l}^r a_i = \min_{i=l}^r b_i$.

Сколько случаев совпадения посчитает робот?

Формат входных данных

В первой строке содержится единственное целое число n ($1 \leq n \leq 200\,000$).

Во второй строке содержатся n целых чисел a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9$) — элементы последовательности a .

В третьей строке содержатся n целых чисел b_1, b_2, \dots, b_n ($-10^9 \leq b_i \leq 10^9$) — элементы последовательности b .

Формат выходных данных

Выведите одно целое число — количество совпадений ответов, которые посчитает робот, то есть для скольких пар выполняется $\max_{i=l}^r a_i = \min_{i=l}^r b_i$.

Примеры

friends.in	friends.out
6 1 2 3 2 1 4 6 7 1 2 3 2	2
3 3 3 3 1 1 1	0

Замечание

Эта задача взята с Codeforces Round 361. Оригинал: <http://codeforces.com/contest/689/problem/D>

Задача С. Безумие и Отвага

Имя входного файла: `heroes.in`
Имя выходного файла: `heroes.out`
Ограничение по времени: 10 секунд
Ограничение по памяти: 512 мегабайт

Многие из нас с детства мечтали создавать компьютерные игры, а для некоторых это даже стало причиной, по которой они начали изучать информатику и программирование. Мишина мечта сбылась, и теперь он работает в известной и уважаемой корпорации «Метель», выпустившей в своё время такие шедевры, как «Искусство войны» и «Звёздное ремесло».

Недавно Миша присоединился к проекту новой ролевой игры «Безумие и отвага». Её ключевой особенностью является возможность на каждом из уровней заново выбирать персонажа для его прохождения.

Перед стартом очередного уровня игроку доступны N героев. Каждый герой характеризуется силой атаки a_i и запасом здоровья b_i . Уровень представляет собой длинную пещеру, содержащую M монстров. Каждый монстр также имеет свою силу атаки c_i и запас здоровья d_i . Зайдя в пещеру, герой сначала сражается с первым монстром, затем, если остаётся жив, сражается со вторым и так далее, пока не погибнет или не дойдёт до конца. Количество жизней героя не восстанавливается между боями, то есть каждую следующую драку он начинает с меньшим запасом здоровья, чем предыдущую.

Бой между монстром и героем состоит в одновременном обмене ударами. Каждый из них, нанося удар, уменьшает запас здоровья противника на величину, равную силе своей атаки. Как только запас здоровья кого-либо из сражающихся становится неположительным, он умирает, и бой прекращается. Обратите внимание, что при такой схеме боя возможна ситуация, когда оба противника погибнут одновременно.

Компания планирует распространять игру бесплатно, получая доход за счёт продажи разнообразных бонусов, реализовать один из которых и поручено Мише. Данный бонус позволяет игроку узнать, сколько монстров убьёт каждый из героев, если игрок выберет именно его для прохождения данного уровня. Так как монстров и героев может быть очень много, Миша столкнулся со сложностями при вычислении необходимых значений и обратился за помощью к вам.

Формат входных данных

В первой строке ввода записаны два целых числа N и M — количество доступных игроку героев и количество монстров в пещере соответственно ($1 \leq N, M \leq 200\,000$).

Следующие N строк описывают героев. Каждая из них содержит два целых числа a_i и b_i , задающих силу атаки и запас здоровья i -го героя ($1 \leq a_i, b_i \leq 10^9$).

Далее следуют M строк, описывающих находящиеся в пещере монстров. Каждое описание состоит из двух целых чисел c_i и d_i , обозначающих параметры i -го монстра ($1 \leq c_i, d_i \leq 200\,000$). Порядок расположения монстров в пещере совпадает с порядком их описания, то есть первым необходимо убить монстра, описанного в строке $N + 2$, а последним — в строке $N + M + 1$.

Формат выходных данных

Выведите N чисел по одному в строке. i -я строка должна содержать ответ для i -го героя.

Примеры

heroes.in	heroes.out
5 3	0
1 2	1
2 2	2
10 10	3
100 10	3
1 100	
2 2	
7 2	
3 20	

Замечание

Бой между первым героем и первым монстром в пещере продлится один ход, после которого герой погибнет, а монстр останется в живых.

Параметры второго героя совпадают с параметрами первого монстра, поэтому они убьют друг друга на первом же ходу боя. Ответ для данного героя равен одному.

Если игрок выберет для прохождения уровня третьего героя, то после боя с первым монстром его запас здоровья будет равен восьми, а после боя со вторым — единице. Для убийства третьего монстра ему необходимо сделать два удара, но он умрёт после первой же его атаки.

У четвёртого героя столько же жизней, сколько и у третьего, но сила атаки гораздо больше, поэтому он пройдёт уровень полностью, хотя и погибнет в последней драке.

Пятый герой обладает минимально возможной силой атаки, но при этом у него большой запас здоровья, поэтому он сможет пройти весь уровень и остаться в живых. После первого боя его запас здоровья будет равен 96, после второго — 82, а в конце игры останется только 22.

Задача D. Фаброзавры-дизайнеры

Имя входного файла: `fabro.in`
Имя выходного файла: `fabro.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Фаброзавры известны своим тонким художественным вкусом и увлечением ландшафтным дизайном. Они живут около очень живописной реки и то и дело перестраивают тропинку, идущую вдоль реки: либо насыпают дополнительной земли, либо срывают то, что есть. Для того, чтобы упростить эти работы, они поделили всю тропинку на горизонтальные участки, пронумерованные от 1 до N , и их переделки устроены всегда одинаково: они выбирают часть дороги от L -ого до R -ого участка (включительно) и изменяют (увеличивают или уменьшают) высоту на всех этих участках на одну и ту же величину (если до начала переделки высоты были разными, то и после переделки они останутся разными).

Поскольку, как уже говорилось, у фаброзавров тонкий художественный вкус, каждый из них считает, что их река лучше всего выглядит с определенной высоты. Поэтому им хочется знать, есть ли поблизости от их дома место на тропинке, где высота на их взгляд оптимальна. Помогите им в этом разобраться.

Формат входных данных

Первая строка входного файла содержит два числа N и M — длину дороги и количество запросов соответственно ($1 \leq N, M \leq 10^5$). На второй строке содержатся N чисел, разделенных пробелами — начальные высоты соответствующих частей дороги; высоты не превосходят 10^4 по модулю. В следующих M строках содержатся запросы по одному на строке.

Запрос $+ L R X$ означает, что высоту частей дороги от L -ой до R -ой (включительно) нужно изменить на X . При этом $1 \leq L \leq R \leq N$, а $|X| \leq 10^4$.

Запрос $? L R X$ означает, что нужно проверить, есть ли между L -ым и R -ым участками (включая эти участки) участок, где дорога проходит точно на высоте X . Гарантируется, что $1 \leq L \leq R \leq N$, а $|X| \leq 10^9$.

Формат выходных данных

На каждый запрос второго типа нужно вывести в выходной файл на отдельной строке одно слово «YES» (без кавычек), если нужный участок существует, и «NO» в противном случае.

Примеры

fabro.in	fabro.out
10 5	NO
0 1 1 3 3 3 2 0 0 1	YES
? 3 5 2	YES
+ 1 4 1	
? 3 5 2	
+ 7 10 2	
? 9 10 3	

Задача Е. Сычи

Имя входного файла: owls.in
Имя выходного файла: owls.out
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Мурка, не ходи, там сыч
На подушке вышит,
Мурка серый, не мурлычь,
Дедушка услышит.
Няня, не горит свеча,
И скребутся мыши.
Я боюсь того сыча,
Для чего он вышит?

Анна Ахматова

Есть n сычей. Известны пары друзей. Три сыча образуют компанию, если все три попарно дружат. Требуется найти количество компаний сычей.

Формат входных данных

Первая строка входных данных содержит два целых числа n и m — количество сычей и дружеских связей соответственно ($1 \leq n, m \leq 3 \cdot 10^5$). Каждая из следующих m строк содержит по два целых числа от 1 до n — номера сычей, которые дружат. Гарантируется, что никакая пара не указана два раза и сыч не дружит сам с собой.

Формат выходных данных

В единственной строке выходного файла выведите количество компаний сычей.

Примеры

owls.in	owls.out
6 6 1 2 2 3 3 1 4 2 3 4 5 1	2