

Задача А. Катый ноль

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

Реализуйте эффективную структуру данных, позволяющую изменять элементы массива и вычислять индекс k -го слева нуля на данном отрезке в массиве.

Формат входных данных

В первой строке вводится одно натуральное число N ($1 \leq N \leq 200\,000$) — количество чисел в массиве. Во второй строке вводятся N чисел от 0 до 100 000 — элементы массива. В третьей строке вводится одно натуральное число M ($1 \leq M \leq 200\,000$) — количество запросов. Каждая из следующих M строк представляет собой описание запроса. Сначала вводится одна буква, кодирующая вид запроса (`s` — вычислить индекс k -го нуля, `u` — обновить значение элемента). Следом за `s` вводится три числа — левый и правый концы отрезка и число k ($1 \leq k \leq N$). Следом за `u` вводятся два числа — номер элемента и его новое значение.

Формат выходных данных

Для каждого запроса `s` выведите результат. Все числа выводите в одну строку через пробел. Если нужного числа нулей на запрашиваемом отрезке нет, выводите -1 для данного запроса.

Примеры

<code>kthzero.in</code>	<code>kthzero.out</code>
5	4
0 0 3 0 2	
3	
u 1 5	
u 1 0	
s 1 5 3	

Замечание

TL для Python 10 секунд

Задача В. Разрезанные 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:

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

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

Замечание

Пояснение к тесту из примера: запросы и результаты.

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
12345	305498	7048017	11694653	1565158	2591019	9471233	570265	13137658	1325095

#	u	v	ans
1	3	9	570265
2	10	1	12345
3	1	2	12345
4	10	10	1325095
5	5	9	570265
6	2	1	12345
7	3	2	305498
8	5	3	1565158

Задача С. Звезды

Имя входного файла:	stars.in
Имя выходного файла:	stars.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вася любит наблюдать за звездами. Но следить за всем небом сразу ему тяжело. Поэтому он наблюдает только за частью пространства, ограниченной кубом размером $n \times n \times n$. Этот куб поделен на маленькие кубики размером $1 \times 1 \times 1$. Во время его наблюдений могут происходить следующие события:

1. В каком-то кубике появляются или исчезают несколько звезд.
2. К нему может заглянуть его друг Петя и поинтересоваться, сколько видно звезд в части пространства, состоящей из нескольких кубиков.

Формат входных данных

Первая строка входного файла содержит натуральное число $1 \leq n \leq 128$. Координаты кубиков — целые числа от 0 до $n - 1$. Далее следуют записи о происшедших событиях по одной в строке. В начале строки записано число m . Если m равно:

- 1, то за ним следуют 4 числа — x, y, z ($0 \leq x, y, z < N$) и k ($-20000 \leq k \leq 20000$) — координаты кубика и величина, на которую в нем изменилось количество видимых звезд;
- 2, то за ним следуют 6 чисел — $x_1, y_1, z_1, x_2, y_2, z_2$ ($0 \leq x_1 \leq x_2 < N, 0 \leq y_1 \leq y_2 < N, 0 \leq z_1 \leq z_2 < N$), которые означают, что Петя попросил подсчитать количество звезд в кубиках (x, y, z) из области: $x_1 \leq x \leq x_2, y_1 \leq y \leq y_2, z_1 \leq z \leq z_2$;
- 3, то это означает, что Васе надоело наблюдать за звездами и отвечать на вопросы Пети. Эта запись встречается во входном файле только один раз и будет последней.

Количество записей во входном файле не больше 100 002.

Формат выходных данных

Для каждого Петиного вопроса выведите искомое количество звезд.

Примеры

stars.in	stars.out
2	0
2 1 1 1 1 1 1	1
1 0 0 0 1	4
1 0 1 0 3	2
2 0 0 0 0 0 0	
2 0 0 0 0 1 0	
1 0 1 0 -2	
2 0 0 0 1 1 1	
3	

Задача D. Перестановки

Имя входного файла: permutation.in
Имя выходного файла: permutation.out
Ограничение по времени: 1.5 секунды
Ограничение по памяти: 256 мегабайт

Вася выписал на доске в каком-то порядке все числа от 1 по N , каждое число ровно по одному разу. Количество чисел оказалось довольно большим, поэтому Вася не может окинуть взглядом все числа. Однако ему надо всё-таки представлять эту последовательность, поэтому он написал программу, которая отвечает на вопрос — сколько среди чисел, стоящих на позициях с x по y , по величине лежат в интервале от k до l . Сделайте то же самое.

Формат входных данных

В первой строке лежит два натуральных числа — $1 \leq N \leq 100\,000$ — количество чисел, которые выписал Вася и $1 \leq M \leq 100\,000$ — количество вопросов, которые Вася хочет задать программе. Во второй строке дано N чисел — последовательность чисел, выписанных Васей. Далее в M строках находятся описания вопросов. Каждая строка содержит четыре целых числа $1 \leq x \leq y \leq N$ и $1 \leq k \leq l \leq N$.

Формат выходных данных

Выведите M строк, каждая должна содержать единственное число — ответ на Васин вопрос.

Примеры

permutation.in	permutation.out
4 2	1
1 2 3 4	3
1 2 2 3	
1 3 1 3	

Задача E. RMQ наоборот

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

Рассмотрим массив $a[1..n]$. Пусть $Q(i, j)$ — ответ на запрос о нахождении минимума среди чисел $a[i], \dots, a[j]$. Вам даны несколько запросов и ответы на них. Восстановите исходный массив.

Формат входных данных

Первая строка входного файла содержит число n — размер массива, и m — число запросов ($1 \leq n, m \leq 100\,000$). Следующие m строк содержат по три целых числа i, j и q , означающих, что $Q(i, j) = q$ ($1 \leq i \leq j \leq n, -2^{31} \leq q \leq 2^{31} - 1$).

Формат выходных данных

Если искомого массива не существует, выведите строку `inconsistent`.

В противном случае в первую строку выходного файла выведите `consistent`. Во вторую строку выходного файла выведите элементы массива. Элементами массива должны быть целые числа в интервале от -2^{31} до $2^{31} - 1$ включительно. Если решений несколько, выведите любое.

Примеры

<code>rmq.in</code>	<code>rmq.out</code>
3 2 1 2 1 2 3 2	<code>consistent</code> 1 2 2
3 3 1 2 1 1 1 2 2 3 2	<code>inconsistent</code>

Задача F. 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 G. Ладьи и прямоугольники

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

У Поликарпа есть шахматная доска размера $n \times m$, на которой расставлены k ладей. Поликарп еще не придумал правила игры, в которую он будет играть. Однако он уже выделил на доске q прямоугольных участков особой стратегической важности, которые должны быть надежно защищены. По мнению Поликарпа, прямоугольный участок доски надежно защищен, если все его свободные клетки бьются ладьями, стоящими на этом участке. Ладьи на остальной части доски на защиту участка не влияют. Расстановка ладей фиксирована и не может быть изменена. Напомним, что ладья бьет все клетки, расположенные с ней на одной вертикали или горизонтали, если между клеткой и ладьей нет других фигур. Помогите Поликарпу определить, все ли стратегически важные участки надежно защищены.

Формат входных данных

В первой строке содержатся четыре целых числа n , m , k и q ($1 \leq n, m \leq 100\,000$, $1 \leq k, q \leq 200\,000$) — размеры доски, количество ладей и количество стратегически важных участков. Будем считать, что клетки доски пронумерованы числами от 1 до n по горизонтали и от 1 до m по вертикали. Следующие k строк содержат пары целых чисел « $x\ y$ », описывающие положение ладей ($1 \leq x \leq n$, $1 \leq y \leq m$). Гарантируется, что все ладьи стоят в разных клетках. Следующие q строк описывают стратегически важные участки четверками чисел « $x_1\ y_1\ x_2\ y_2$ » ($1 \leq x_1 \leq x_2 \leq n$, $1 \leq y_1 \leq y_2 \leq m$). Соответствующий прямоугольный участок состоит из клеток (x, y) , для которых $x_1 \leq x \leq x_2$, $y_1 \leq y \leq y_2$. Стратегически важные участки могут пересекаться или совпадать.

Формат выходных данных

Выведите q строк. Для каждого стратегически важного участка выведите «YES», если он надежно защищен, и «NO» в противном случае.

Примеры

<code>rooks.in</code>	<code>rooks.out</code>
4 3 3 3 1 1 3 2 2 3 2 3 2 3 2 1 3 3 1 2 2 3	YES YES NO
1 1 1 1 1 1 1 1 1 1	YES