



RVQ. Range Variation Query

Имя входного файла: rvq.in
Имя выходного файла: rvq.out

Последовательность чисел a_n задана следующей формулой:

$$a_n = n^2 \bmod 12345 + n^3 \bmod 23456.$$

Реализуйте структуру данных, которая позволяет обрабатывать запросы следующего вида:

- найти разность между максимальным и минимальным значением среди элементов a_i, a_{i+1}, \dots, a_j ;
- присвоить элементу a_i новое значение j .

Формат входного файла

Первая строка входного файла содержит натуральное число k — количество запросов ($k \leq 100\,000$). Следующие k строк содержат запросы, по одному на строке. Запрос номер i описывается двумя целыми числами x_i, y_i .

Если $x_i > 0$, то требуется найти разность между максимальным и минимальным значением среди элементов $a_{x_i} \dots a_{y_i}$. При этом $1 \leq x_i \leq y_i \leq 100\,000$.

Если $x_i < 0$, то требуется присвоить элементу $a_{|x_i|}$ значение y_i . При этом $-100\,000 \leq x_i \leq -1$ и $|y_i| \leq 100\,000$.

Формат выходного файла

Для каждого запроса первого типа в выходной файл требуется вывести одну строку, содержащую разность между максимальным и минимальным значением на соответствующем отрезке.

Пример

rvq.in	rvq.out
7	34
1 3	68
2 4	250
-2 -100	234
1 5	1
8 9	
-3 -101	
2 3	

Billboard. Доска объявлений

Имя входного файла: billboard.in
Имя выходного файла: billboard.out

На входе в гостиницу “Русский капитал” висит доска объявлений высоты h и ширины w . На доске вывешивают расписание на день, списки детей, идущих в театр, и другую полезную информацию.

28 декабря доска была свободна, затем на неё одно за другим стали вывешивать объявления.

i -е объявление — это полоска бумаги единичной высоты и ширины w_i . Когда на доску вывешивается новое объявление, то оно крепится как можно выше. Если есть несколько способов это сделать, оно крепится как можно левее. Если для объявления не хватает места, оно вообще не вывешивается.

Ваша задача — для каждого объявления определить номер ряда, где оно будет закреплено.

Формат входного файла

В первой строке записаны целые числа h, w, n ($1 \leq h, w \leq 10^9$; $1 \leq n \leq 200\,000$) — размеры доски и количество объявлений.

В i -й из следующих n строк записано число w_i ($1 \leq w_i \leq 10^9$) — ширина i -го объявления.

Формат выходного файла

Для каждого объявления (в том порядке, в котором они заданы на входе) выведите номер ряда, где оно будет закреплено. Ряды занумерованы числами от 1 до h сверху вниз. Если объявление закрепить нельзя, выведите “-1”.

Пример

billboard.in	billboard.out
3 5 5	1
2	2
4	1
3	3
3	-1
3	



Sparse. Разреженные таблицы

Имя входного файла: `sparse.in`

Имя выходного файла: `sparse.out`

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 входного файла

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 ($0 \leq a_1 < 16714589$) — 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:

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

где ans_i — ответ на запрос номер i .

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

Формат выходного файла

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