

Задача А. Range Variation Query

Имя входного файла: `rvq.in`
Имя выходного файла: `rvq.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В начальный момент времени последовательность a_n задана следующей формулой: $a_n = n^2 \bmod 12345 + n^3 \bmod 23456$.

Требуется много раз отвечать на запросы следующего вида:

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

Задача В. Разреженные таблицы

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

Sparse

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 ($1 \leq n \leq 10^5, m \leq 10^7$) 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:

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

где ans_i — ответ на запрос номер i .

Формат выходного файла

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

Пример

sparse.in	sparse.out
10 8 12345	5 3 1565158
3 9	

Задача С. Отсортированные запросы на минимум

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

Дан массив, состоящий из N целых чисел a_0, a_1, \dots, a_{N-1} .

Также дано Q запросов на минимум $[l_i, r_i]$, для каждого из которых необходимо найти минимум на соответствующем отрезке массива. Известно, что все левые границы запросов расположены по неубыванию, равно как и правые. Иными словами $l_i \leq l_{i+1}, r_i \leq r_{i+1}$ для всех i от 1 до $Q-1$.

Формат входного файла

Первая строка входного файла содержит целое число $1 \leq N \leq 300\,000$. Следующая строка содержит N чисел, не превосходящих по модулю 2^{30} , которые являются элементами массива. Третья строка содержит целое число $1 \leq Q \leq 500\,000$, количество запросов. Следующие Q строк содержат числа $0 \leq l_i \leq r_i \leq N-1$, удовлетворяющие описанным выше условиям.

Формат выходного файла

Для каждого запроса выведите в отдельной строке ответ на него.

Пример

sortedrmq.in	sortedrmq.out
6	-4
-4 4 -2 3 -4 8	-2
5	3
0 1	-4
1 2	8
3 3	
4 5	
5 5	

Задача D. Окна

Имя входного файла: windows.in
Имя выходного файла: windows.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

На экране расположены прямоугольные окна, возможно, каким-то образом перекрывающиеся (со сторонами, параллельными осям координат). Вам необходимо найти точку, которая покрыта наибольшим числом из них.

Формат входного файла

В первой строке входного файла записано число окон n ($1 \leq n \leq 50\,000$). Следующие n строк содержат координаты окон $x_{(1,i)}, y_{(1,i)}, x_{(2,i)}, y_{(2,i)}$, где $(x_{(1,i)}, y_{(1,i)})$ — координаты левого верхнего угла i -го окна, а $(x_{(2,i)}, y_{(2,i)})$ — правого нижнего (на экране компьютера y растет сверху вниз, а x — слева направо). Все координаты — целые числа, по модулю не превосходящие $2 \cdot 10^5$.

Формат выходного файла

В первой строке выходного файла выведите максимальное число окон, покрывающих какую-либо из точек в данной конфигурации. Во второй строке выведите два целых числа, разделенные пробелом — координаты точки, покрытой максимальным числом окон. Окна считаются замкнутыми, т. е. покрывающими свои граничные точки.

Пример

windows.in	windows.out
2	2
0 0 3 3	3 2
1 1 4 4	