RVQ. Range Variation Query

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

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 $a_i, a_{i+1}, ..., a_j$;
- присвоить элементу a_i значение j.

Формат входного файла

Первая строка входного файла содержит натуральное число k — количество запросов ($k \leqslant 100\,000$). 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 $a_{x_i}...a_{y_i}$. При этом $1 \leqslant x_i \leqslant y_i \leqslant 100\,000$.

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

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 1 3 2 4 -2 -100 1 5 8 9 -3 -101 2 3	34 68 250 234 1

Floor4. Четвертый этаж

Имя входного файла: `floor4.in`
Имя выходного файла: `floor4.out`

Знаете ли вы, почему четвертый этаж заперт и там не останавливается лифт? Потому что на самом деле четвертый, запертый, этаж, где не останавливается лифт, содержит бесконечное количество комнат, пронумерованных натуральными числами. На этот этаж регулярно приезжают дети, каждый из которых заранее выбрал, в какую комнату он хочет заселиться. Если выбранная комната оказывается свободна, то ребенок занимает ее, в противном случае он занимает первую свободную комнату с бо́льшим номером.

Кроме того, некоторые дети уезжают в середине смены. Сразу после отъезда ребенка его комната становится доступна для заселения следующего.

Промоделируйте работу преподавателей, ответственных за четвертый этаж, и научитесь быстро сообщать приезжающим детям, какую комнату им следует занимать.

Формат входного файла

Первая строка входного файла содержит натуральное число n — количество прибытий и отъездов, происходящих в течение смены ($n \leqslant 100\,000$).

Следующие n строк содержат информацию об ЛКШатах. Число $a > 0$ обозначает, что приехал школьник, желающий занять комнату номер a ($a \leqslant 100\,000$). Число $a < 0$ обозначает, что с комнаты номер $|a|$ уехал школьник. (Гарантируется, что эта комната не была пуста).

Формат выходного файла

Для каждого приезжающего школьника выведите одно натуральное число — номер комнаты, в которую он поселится.

Пример

floor4.in	floor4.out
6 5 5 5 -6 5 5	5 6 7 6 8

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 ($1 \leqslant n \leqslant 10^5, m \leqslant 10^7$) и a_1 ($0 \leqslant a_1 < 16714589$) — количество элементов в массиве, количество запросов и первый элемент массива соответственно. Вторая строка содержит два натуральных числа u_1 и v_1 ($1 \leqslant u_1, v_1 \leqslant n$) — первый запрос.

Элементы $a_2, a_3, \ldots, a_n$ задаются следующей формулой:

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

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:

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

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 3 9	5 3 1565158

Docs. Предъявите документы!

Имя входного файла: `docs.in`
Имя выходного файла: `docs.out`

При рождении человеку выдаётся N документов, пронумерованных от 1 до N. Документ с номером i характеризуется своей важностью A_i и стоимостью B_i. В течение жизни человек принимает участие в M важных событиях, для участия в каждом из которых ему требуется отдать один из своих документов. Более того, для участия в событии с номером j необходимо отдать документ с важностью не меньше C_j и не больше D_j. При этом, естественно, человек каждый раз отдаёт самый дешёвый документ из подходящих. Помогите человеку прожить жизнь, поучаствовав во всех важных событиях.

Формат входного файла

Во входном файле содержится число N $(1 \leqslant N \leqslant 10^5)$, затем N пар чисел A_i и B_i $(1 \leqslant A_i, B_i \leqslant 10^9)$, затем число M $(1 \leqslant M \leqslant 10^5)$, затем M пар чисел C_j и D_j $(1 \leqslant C_j \leqslant D_j \leqslant 10^9)$. Стоимости всех документов различны.

Формат выходного файла

В выходной файл запишите M чисел — номера документов, которые необходимо отдать, j-е число обозначает номер документа, отдаваемый на j-м событии. Числа в строке разделяйте пробелом. В случае если жизнь прожить невозможно, выведите в выходной файл одно слово `BOTVA`.

Пример

docs.in	docs.out
3 5 99 1 100 3 50 2 2 5 1 4	3 2
3 5 99 1 100 3 50 2 2 5 3 4	BOTVA