Kthmax. K-ый максимум

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

Напишите программу, реализующую декартово дерево, позволяющую добавлять и удалять элементы, а также находить k-й максимум.

Формат входного файла

Первая строка входного файла содержит натуральное число n — количество команд ($n \leqslant 100\,000$). Последующие n строк содержат по одной команде каждая. Команда записывается в виде двух чисел c_i и k_i — тип и аргумент команды соответственно ($|k_i| \leqslant 10^9$). Поддерживаемые команды:

- $+1$: Добавить элемент с ключом k_i.
- 0: Найти и вывести k_i-й максимум.
- -1: Удалить элемент с ключом k_i.

Гарантируется, что в процессе работы в структуре не требуется хранить элементы с равными ключами или удалять несуществующие элементы. Также гарантируется, что при запросе k_i-го максимума, он существует.

Формат выходного файла

Для каждой команды нулевого типа в выходной файл должна быть выведена строка, содержащая единственное число — k_i-й максимум.

Пример

kthmax.in	kthmax.out
8 +1 5 +1 3 +1 7 0 2 -1 3 0 1 1 10 0 3	5 7 5

RMQ. RMQ Наоборот (*)

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

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

Формат входного файла

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

Формат выходного файла

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

rmq.in	rmq.out
3 2 1 2 1 2 3 2	consistent 1 2 3
3 3 1 2 1 1 1 2 2 3 2	inconsistent