

Задача А. Простое двоичное дерево поиска

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Реализуйте просто двоичное дерево поиска.

Формат входных данных

Входной файл содержит описание операций с деревом, их количество не превышает 100. В каждой строке находится одна из следующих операций:

- **insert** x — добавить в дерево ключ x . Если ключ x есть в дереве, то ничего делать не надо;
- **delete** x — удалить из дерева ключ x . Если ключа x в дереве нет, то ничего делать не надо;
- **exists** x — если ключ x есть в дереве выведите «**true**», если нет «**false**»;
- **next** x — выведите минимальный элемент в дереве, строго больший x , или «**none**» если такого нет;
- **prev** x — выведите максимальный элемент в дереве, строго меньший x , или «**none**» если такого нет.

В дерево помещаются и извлекаются только целые числа, не превышающие по модулю 10^9 .

Формат выходных данных

Выведите последовательно результат выполнения всех операций **exists**, **next**, **prev**. Следуйте формату выходного файла из примера.

Пример

стандартный ввод	стандартный вывод
insert 2	true
insert 5	false
insert 3	5
exists 2	3
exists 4	none
next 4	3
prev 4	
delete 5	
next 4	
prev 4	

Задача В. Сбалансированное двоичное дерево поиска

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Реализуйте сбалансированное двоичное дерево поиска.

Формат входных данных

Входной файл содержит описание операций с деревом, их количество не превышает 10^5 . В каждой строке находится одна из следующих операций:

- **insert** x — добавить в дерево ключ x . Если ключ x есть в дереве, то ничего делать не надо;
- **delete** x — удалить из дерева ключ x . Если ключа x в дереве нет, то ничего делать не надо;
- **exists** x — если ключ x есть в дереве выведите «**true**», если нет «**false**»;
- **next** x — выведите минимальный элемент в дереве, строго больший x , или «**none**» если такого нет;
- **prev** x — выведите максимальный элемент в дереве, строго меньший x , или «**none**» если такого нет.

В дерево помещаются и извлекаются только целые числа, не превышающие по модулю 10^9 .

Формат выходных данных

Выведите последовательно результат выполнения всех операций **exists**, **next**, **prev**. Следуйте формату выходного файла из примера.

Пример

стандартный ввод	стандартный вывод
insert 2	true
insert 5	false
insert 3	5
exists 2	3
exists 4	none
next 4	3
prev 4	
delete 5	
next 4	
prev 4	

Задача С. Переместить в начало

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 6 секунд
Ограничение по памяти: 512 мегабайт

Вам дан массив $a_1 = 1, a_2 = 2, \dots, a_n = n$ и последовательность операций: переместить элементы с l_i по r_i в начало массива. Например, для массива $2, 3, 6, 1, 5, 4$, после операции $(2, 4)$ новый порядок будет $3, 6, 1, 2, 5, 4$. А после применения операции $(3, 4)$ порядок элементов в массиве будет $1, 2, 3, 6, 5, 4$.

Выведите порядок элементов в массиве после выполнения всех операций.

Формат входных данных

В первой строке входного файла указаны числа n и m ($2 \leq n \leq 100\,000$, $1 \leq m \leq 100\,000$) — число элементов в массиве и число операций. Следующие m строк содержат операции в виде двух целых чисел: l_i и r_i ($1 \leq l_i \leq r_i \leq n$).

Формат выходных данных

Выведите n целых чисел — порядок элементов в массиве после применения всех операций.

Пример

стандартный ввод	стандартный вывод
6 3 2 4 3 5 2 2	1 4 5 2 3 6

Задача D. K -й максимум

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

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

Формат входных данных

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

- $+1$ (или просто 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 выходных данных

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

Пример

стандартный ввод	стандартный вывод
11	7
+1 5	5
+1 3	3
+1 7	10
0 1	7
0 2	3
0 3	
-1 5	
+1 10	
0 1	
0 2	
0 3	

Задача Е. Декартово дерево

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вам даны пары чисел (a_i, b_i) . Необходимо построить декартово дерево, такое что i -я вершина имеет ключи (a_i, b_i) , вершины с ключом a_i образуют бинарное дерево поиска, а вершины с ключом b_i образуют кучу.

Формат входных данных

В первой строке записано число N — количество пар. Далее следует N ($1 \leq N \leq 300\,000$) пар (a_i, b_i) . Для всех пар $|a_i|, |b_i| \leq 1\,000\,000$. $a_i \neq a_j$ и $b_i \neq b_j$ для всех $i \neq j$.

Формат выходных данных

Если декартово дерево с таким набором ключей построить возможно, выведите в первой строке «YES», в противном случае выведите «NO». В случае ответа «YES» выведите N строк, каждая из которых должна описывать вершину. Описание вершины состоит из трёх чисел: номера предка, номера левого сына и номера правого сына. Если у вершины отсутствует предок или какой либо из сыновей, выведите на его месте число 0.

Если подходящих деревьев несколько, выведите любое.

Пример

стандартный ввод	стандартный вывод
7	YES
5 4	2 3 6
2 2	0 5 1
3 9	1 0 7
0 5	5 0 0
1 3	2 4 0
6 6	1 0 0
4 11	3 0 0

Задача F. Добавление ключей

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вы работаете в компании Макрохард и вас попросили реализовать структуру данных, которая будет хранить множество целых ключей.

Будем считать, что ключи хранятся в бесконечном массиве A , проиндексированном с 1, исходно все его ячейки пусты. Структура данных должна поддерживать следующую операцию:

$\text{Insert}(L, K)$, где L — позиция в массиве, а K — некоторое положительное целое число.

Операция должна выполняться следующим образом:

- Если ячейка $A[L]$ пуста, присвоить $A[L] \leftarrow K$.
- Если $A[L]$ непуста, выполнить $\text{Insert}(L + 1, A[L])$ и затем присвоить $A[L] \leftarrow K$.

По заданным N целым числам L_1, L_2, \dots, L_N выведите массив после выполнения последовательности операций:

$\text{Insert}(L_1, 1) \text{ Insert}(L_2, 2) \dots \text{Insert}(L_N, N)$

Формат входных данных

Первая строка входного файла содержит числа N — количество операций Insert , которое следует выполнить и M — максимальную позицию, которая используется в операциях Insert ($1 \leq N \leq 131\,072$, $1 \leq M \leq 131\,072$).

Следующая строка содержит N целых чисел L_i , которые описывают операции Insert , которые следует выполнить ($1 \leq L_i \leq M$).

Формат выходных данных

Выведите содержимое массива после выполнения всех сделанных операций Insert . На первой строке выведите W — номер максимальной непустой ячейки в массиве. Затем выведите W целых чисел — $A[1], A[2], \dots, A[W]$. Выводите нули для пустых ячеек.

Пример

стандартный ввод	стандартный вывод
5 4 3 3 4 1 3	6 4 0 5 2 3 1

Задача G. И снова сумма

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Реализуйте структуру данных, которая поддерживает множество S целых чисел, с которым разрешается производить следующие операции:

- $\text{add}(i)$ — добавить в множество S число i (если он там уже есть, то множество не меняется);
- $\text{sum}(l, r)$ — вывести сумму всех элементов x из S , которые удовлетворяют неравенству $l \leq x \leq r$.

Формат входных данных

Исходно множество S пусто. Первая строка входного файла содержит n — количество операций ($1 \leq n \leq 300\,000$). Следующие n строк содержат операции. Каждая операция имеет вид либо «+ i », либо «? l r ». Операция «? l r » задает запрос $\text{sum}(l, r)$.

Если операция «+ i » идет во входном файле в начале или после другой операции «+», то она задает операцию $\text{add}(i)$. Если же она идет после запроса «?», и результат этого запроса был y , то выполняется операция $\text{add}((i + y) \bmod 10^9)$.

Во всех запросах и операциях добавления параметры лежат в интервале от 0 до 10^9 .

Формат выходных данных

Для каждого запроса выведите одно число — ответ на запрос.

Пример

стандартный ввод	стандартный вывод
6	3
+ 1	7
+ 3	
+ 3	
? 2 4	
+ 1	
? 2 4	