Содержание

У вас не получается читать/выводить данные?
Воспользуйтесь примерами (`c++`) (`python`).

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

Must have

Задача 11A. Простейшее BST [0.3 (3), 256]

В этой задаче вам нужно написать простейшее BST по явному ключу и отвечать им на запросы:

- `+ x` – добавить в дерево `x` (если `x` уже есть, ничего не делать).
- `> x` – вернуть минимальный элемент больше `x` или 0, если таких нет.

Формат входных данных

В каждой строке содержится один запрос.
Все x целые от 1 до 10^9, количество запросов от 1 до 300 000.
Гарантируется, что все x выбраны равномерным распределением.

Формат выходных данных

Для каждого запроса вида «`> x`» выведите в отдельной строке ответ.

Пример

stdin	stdout
+ 1 + 3 + 3 > 1 > 2 > 3 + 2 > 1	3 3 0 2

Замечание

Случайные данные! Не нужно ничего специально балансировать.

Как именно делать `lowerbound` разобранно на практике.
В этой задаче нужно написать своё дерево, уже готовыми пользоваться нельзя.

Задача 11B. Простейший неявный Ключ [0.4 (4), 256]

Изначально есть пустой массив. Вам нужно обрабатывать запросы вида
`i x` — добавить после i-го элемента x $(0 \leqslant i \leqslant n)$, где n текущая длина массива.
В конце после всех добавлений нужно вывести полученный массив.

Формат входных данных

q $(1 \leqslant q \leqslant 100\,000)$ строк, на каждой запрос «`i x`» —
пара целых чисел $(0 \leqslant i \leqslant n,\ 1 \leqslant x < 100\,000)$.

Формат выходных данных

Выведите q целых чисел — полученный массив.

Примеры

stdin	stdout
0 1 0 2 0 3	3 2 1
0 1 1 2 2 3	1 2 3
0 1 1 2 1 3 0 4	4 1 3 2

Замечание

Случайные данные! Не нужно ничего специально балансировать.

Вам нужно в каждой вершине поддерживать размер поддерева.
В этой задаче вам нужно написать BST по **не**явному ключу.
В этой задаче нужно написать своё дерево, уже готовыми пользоваться нельзя.

Задача 11C. Сумма [0.4 (4), 256]

Дан массив из N элементов, нужно научиться находить сумму чисел на отрезке.

Формат входных данных

Первая строка содержит два целых числа N и K — число чисел в массиве и количество запросов. ($1 \le N \le 100\,000$), ($0 \le K \le 100\,000$). Следующие K строк содержат запросы

- "`A i x`" — присвоить i-му элементу массива значение x ($1 \le i \le n$, $0 \le x \le 10^9$)
- "`Q l r`" — найти сумму чисел в массиве на позициях от l до r. ($1 \le l \le r \le n$)

Изначально в массиве живут нули.

Формат выходных данных

На каждый запрос вида Q l r нужно вывести единственное число — сумму на отрезке.

Примеры

stdin	stdout
5 9 A 2 2 A 3 1 A 4 2 Q 1 1 Q 2 2 Q 3 3 Q 4 4 Q 5 5 Q 1 5	0 2 1 2 0 5

Замечание

Обыкновенное дерево отрезков. Простейшее.
В этой задаче нужно написать своё дерево, уже готовыми пользоваться нельзя.

Задачи здорового человека

Задача 11D. Простейший полуявный ключ [0.3 (3), 256]

В этой задаче вам нужно написать BST по **явному** ключу и отвечать им на запросы:

- `+ x` – добавить в дерево `x` (если `x` уже есть, ничего не делать).
- `? k` – вернуть k-й по возрастанию элемент.

Формат входных данных

В каждой строке содержится один запрос.
Все x целые от 1 до 10^9, количество запросов от 1 до 300 000.
Гарантируется, что все x выбраны равномерным распределением.
В запросах «`? k`», число k от 1 до количества элементов в дереве.

Формат выходных данных

Для каждого запроса вида «`? k`» выведите в отдельной строке ответ.

Пример

stdin	stdout
+ 1 + 4 + 3 + 3 ? 1 ? 2 ? 3 + 2 ? 3	1 3 4 3

Подсказка по решению

Случайные данные! Не нужно ничего специально балансировать.

Вам нужно в каждой вершине поддерживать размер поддерева.
В этой задаче вам нужно написать BST по **не**явному ключу.
В этой задаче нужно написать своё дерево, уже готовыми пользоваться нельзя.

Задача 11E. Двоичное дерево поиска [0.5 (3.3), 256]

Реализуйте сбалансированное двоичное дерево поиска.
Или воспользуйтесь любой стандартной структурой из `C++::STL`.
Или попытайтесь запихать квадрат.

Формат входных данных

Входной файл содержит описание одной или нескольких операций с деревом.
Операций не больше 10^5. Все числа целые от -10^5 до 10^9.
В каждой строке находится одна из следующих операций:

- `insert` x — добавить в дерево ключ x.
  Если ключ x в дереве уже есть, 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, «`true`», иначе «`false`»
- `next` x — минимальный элемент в дереве, $> x$, или «`none`», если такого нет.
- `prev` x — максимальный элемент в дереве, $< x$, или «`none`», если такого нет.

Формат выходных данных

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

Примеры

stdin	stdout
insert 2 insert 5 insert 3 exists 2 exists 4 next 4 prev 4 delete 5 next 4 prev 4	true false 5 3 none 3

Подсказка по решению

В любом языке есть стандартная структура, которая умеет всё описанное.
В этой задаче вам нужно научиться пользоваться возможностями вашего языка.

`c++`: `set<int>`
`java/kotlin`: `TreeSet<Integer>`
`python`: `sortedcontainers.SortedSet` [sortedset]

В этой задаче **не** нужно писать своё дерево.

Задача 11F. K-ый максимум [0.5 (4.5), 256]

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 ($n \leqslant 100\,000$). 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:

- $+1$ (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

stdin	stdout
11 +1 5 +1 3 +1 7 0 1 0 2 0 3 -1 5 +1 10 0 1 0 2 0 3	7 5 3 10 7 3

Подсказка по решению

Напишите своё BST.
Можно любое: avl, treap, rb, splay, что больше нравится. На парах вы проходили avl.

Для искателей острых ощущений

Задача 11G. Неявный Ключ [1.5 (7), 256]

Научитесь быстро делать две операции с массивом:
- `add i x` — добавить после i-го элемента x $(0 \leqslant i \leqslant n)$
- `del i` — удалить i-й элемент $(1 \leqslant i \leqslant n)$

Формат входных данных

На первой строке n_0 и m $(1 \leqslant n_0, m \leqslant 10^5)$ — длина исходного массива и количество запросов. На второй строке n_0 целых чисел от 0 до $10^9 - 1$ — исходный массив. Далее m строк, содержащие запросы. Гарантируется, что запросы корректны: например, если просят удалить i-й элемент, он точно есть.

Формат выходных данных

Выведите конечное состояние массива. На первой строке количество элементов, на второй строке сам массив.

Примеры

stdin	stdout
3 4 1 2 3 del 3 add 0 9 add 3 8 del 2	3 9 2 8

Подсказка по решению

Напишите своё BST по неявному ключу.

Можно любое: avl, treap, rb, splay, что больше нравится. На парах вы проходили avl.

Задача 11H. И снова сумма... [2 (8), 256]

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

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

Формат входных данных

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

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

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

stdin	stdout
6 + 1 + 3 + 3 ? 2 4 + 1 ? 2 4	3 7

Подсказка по решению

Достаточно взять уже написанное BST и добавить к нему нужную функцию.

Запросы так странно генерятся, чтобы заставить вас обрабатывать их в online.

Задача 11I. Вставка ключевых значений [2.5 (9), 256]

Вас наняла на работу компания MacroHard, чтобы вы разработали новую структуру данных для хранения целых ключевых значений.

Эта структура выглядит как массив A бесконечной длины, ячейки которого нумеруются с единицы. Изначально все ячейки пусты. Единственная операция, которую необходимо поддерживать — это операция `Insert(L,K)`, где L — положение в массиве, а K — некоторое положительное целое ключевое значение. Операция выполняется следующим образом:

- Если ячейка `A[L]` пуста, то присвоить `A[L] := K`.
- Иначе выполнить `Insert(L+1,A[L])`, а затем присвоить `A[L] := K`.

По заданной последовательности из N целых чисел $L_1, L_2, \ldots, L_N$ вам необходимо вывести содержимое этого массива после выполнения следующей последовательности операций:

```
Insert(L_1, 1)
Insert(L_2, 2)
...
Insert(L_N, N)
```

Формат входных данных

В первой строке входного файла содержится N (число операций `Insert`) и M (максимальный номер позиции, которую можно использовать в операции `Insert`). ($1 \leqslant N, M \leqslant 131\,072$).

В следующей строке даны N целых чисел L_i ($1 \leqslant L_i \leqslant M$).

Формат выходных данных

Выведите содержимое массива после выполнения данной последовательности операций `Insert`. На первой строке выведите W — номер последней несвободной позиции в массиве. Далее выведите W целых чисел — $A[1], A[2], \ldots, A[W]$. Для пустых ячеек выводите нули.

Пример

stdin	stdout
5 4 3 3 4 1 3	6 4 0 5 2 3 1