

Содержание

Must have	2
Задача 11А. К-ый максимум [0.2 sec, 256 mb]	2
Обязательные задачи	3
Задача 11В. И снова сумма... [2 sec, 256 mb]	3
Задача 11С. Неявный Ключ [0.6 sec, 256 mb]	4
Задача 11D. Range Minimum Query [0.8 sec, 256 mb]	5
Дополнительные задачи	6
Задача 11Е. Перестановки [4 sec, 256 mb]	6

Вы не умеете читать/выводить данные, открывать файлы? Воспользуйтесь **примерами**.

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) **переопределение стандартного аллокатора** ускорит вашу программу.

Обратите внимание на компилятор GNU C++11 5.1.0 (TDM-GCC-64) inc, который позволяет пользоваться **дополнительной библиотекой**. Под ним можно сдать **вот это**.

Must have

Задача 11А. К-ый максимум [0.2 sec, 256 mb]

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

stdin	stdout
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	

Замечание

Напишите сюда, пожалуйста, именно **AVL дерево**.

Круто, если напишите честное удаление, но гораздо проще удалять лениво.

Из лекции мы знаем, что вращений будет $\mathcal{O}(1) \Rightarrow$ оптимизировать вращение не нужно.

Из допдз мы также знаем, что высота поменяется в среднем у $\mathcal{O}(1)$ вершин.

Обязательные задачи

Задача 11В. И снова сумма... [2 сек, 256 mb]

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

- $add(i)$ — добавить в множество S число i , если i там уже есть, S не меняется;
- $sum(l, r)$ — вывести сумму всех элементов x из $S: l \leq x \leq r$.

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 $sum(l, r)$.

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	3
+ 1	7
+ 3	
+ 3	
? 2 4	
+ 1	
? 2 4	

Задача 11С. Неявный Ключ [0.6 sec, 256 mb]

Научитесь быстро делать две операции с массивом:

- `add i x` — добавить после i -го элемента x ($0 \leq i \leq n$)
- `del i` — удалить i -й элемент ($1 \leq i \leq n$)

Формат входных данных

На первой строке n_0 и m ($1 \leq n_0, m \leq 10^5$) — 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	3
1 2 3	9 2 8
del 3	
add 0 9	
add 3 8	
del 2	

Задача 11D. Range Minimum Query [0.8 sec, 256 mb]

Компания *Giggle* открывает свой новый офис в Судиславле, и вы приглашены на собеседование. Ваша задача — решить поставленную задачу.

Вам нужно создать структуру данных, которая представляет из себя массив целых чисел. Изначально массив пуст. Вам нужно поддерживать две операции:

- запрос: «? i j » — возвращает минимальный элемент между i -ым и j -м, включительно;
- изменение: «+ i x » — добавить элемент x после i -го элемента списка.
Если $i = 0$, то элемент добавляется в начало массива.

Конечно, эта структура должна быть достаточно хорошей.

Формат входных данных

Первая строка входного файла содержит единственное целое число n — число операций над массивом ($1 \leq n \leq 200\,000$). Следующие n строк описывают сами операции. Все операции добавления являются корректными. Все числа, хранящиеся в массиве, по модулю не превосходят 10^9 .

Формат выходных данных

Для каждой операции в отдельной строке выведите её результат.

Примеры

stdin	stdout
8	4
+ 0 5	3
+ 1 3	1
+ 1 4	
? 1 2	
+ 0 2	
? 2 4	
+ 4 1	
? 3 5	

Дополнительные задачи

Задача 11Е. Перестановки [4 сек, 256 mb]

Рассмотрим циклический алфавит, состоящий из первых десяти букв английского алфавита. Циклическим он называется потому, что следующей буквой за 'a' является буква 'b', за 'b' — 'c' и так далее, и при этом следующей за буквой 'j' является буква 'a'.

Вам дана строка S , состоящая из букв этого циклического алфавита. Вы должны обрабатывать запросы трёх типов:

- Развернуть подстроку строки S с L -го по R -й символ включительно.
- В подстроке S с L -го по R -й символ заменить каждый символ на D -й следующий символ в циклическом алфавите.
- Для подстроки S с L -го по R -й символ найти количество различных перестановок символов этой подстроки по модулю $10^9 + 7$.

Формат входных данных

Первая строка входного файла содержит одно число N ($1 \leq N \leq 10^5$) — длину строки S . Во второй строке содержится сама строка S , состоящая из строчных букв английского алфавита от 'a' до 'j'. Третья строка содержит число M ($1 \leq M \leq 10^5$) — количество запросов. Затем идут M строк, содержащие описания запросов трех типов:

- $-1 \ L \ R$ ($1 \leq L \leq R \leq N$) — разворот подстроки.
- $0 \ L \ R \ D$ ($1 \leq L \leq R \leq N, 0 < D \leq N$) — замена символов в подстроке.
- $1 \ L \ R$ ($1 \leq L \leq R \leq N$) — количество различных перестановок подстроки $[L, R]$ строки S .

Формат выходных данных

На каждый запрос типа " $1 \ L \ R$ " выведите ответ по модулю $10^9 + 7$ в отдельной строке.

Пример

stdin	stdout
6	180
abcabc	
3	
-1 1 6	
0 1 3 1	
1 1 6	

Замечание

D -м следующим символом за x называется символ, получающийся путем D -кратного взятия следующего за x символа в циклическом алфавите.