Содержание

В некоторых задачах большой ввод и вывод. Имеет смысл пользоваться супер быстрым вводом-выводом: `http://acm.math.spbu.ru/~sk1/algo/input-output/`

Задачи

Задача A. И снова сумма... [3 sec, 256 mb]

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

sum2.in	sum2.out
6 + 1 + 3 + 3 ? 2 4 + 1 ? 2 4	3 7

Задача B. K-ый максимум [0.5 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 ($n \leqslant 100\,000$). 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

kthmax.in	kthmax.out
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

Задача C. Неявный Ключ [1.5 sec, 256 mb]

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

implicitkey.in	implicitkey.out
3 4 1 2 3 del 3 add 0 9 add 3 8 del 2	3 9 2 8

Задача D. Range Minimum Query [1.5 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 ($1 \leqslant n \leqslant 200\,000$). 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

rmq.in	rmq.out
8 + 0 5 + 1 3 + 1 4 ? 1 2 + 0 2 ? 2 4 + 4 1 ? 3 5	4 3 1

Задача E. Вперёд! [1 sec, 256 mb]

Капрал Дукар любит раздавать приказы своей роте. Самый любимый его приказ — "Вперёд!". Капрал строит солдат в ряд и отдаёт некоторое количество приказов, каждый из них звучит так: "Рядовые с l_i по l_j — вперёд!"

Перед тем, как Дукар отдал первый приказ, солдаты были пронумерованы от 1 до n, слева направо. Услышав приказ "Рядовые с l_i по l_j — вперёд!", солдаты, стоящие на местах с l_i по l_j включительно, продвигаются в начало ряда, в том же порядке, в котором были.

Например, если в какой-то момент солдаты стоят в порядке $1, 3, 6, 2, 5, 4$, то после приказа "Рядовые с 2 по 3 — вперёд!", порядок будет таким: $3, 6, 1, 2, 5, 4$. А если потом Капрал вышлет вперёд солдат с 3 по 4, то порядок будет уже таким: $1, 2, 3, 6, 5, 4$.

Вам дана последовательность из приказов Капрала. Найдите порядок, в котором будут стоять солдаты после исполнения всех приказов.

Формат входных данных

В первой строке входного файла указаны числа n и m ($2 \leqslant n \leqslant 100\,000$, $1 \leqslant m \leqslant 100\,000$) — число солдат и число приказов. Следующие m строк содержат приказы в виде двух целых чисел: l_i и r_i ($1 \leqslant l_i \leqslant r_i \leqslant n$).

Формат выходных данных

Выведите в выходной файл n целых чисел — порядок, в котором будут стоять солдаты после исполнения всех приказов.

Пример

movetofront.in	movetofront.out
6 3 2 4 3 5 2 2	1 4 5 2 3 6

Бонус

Задача F. Вставка ключевых значений [2 sec, 256 mb]

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 $A[L] := K$.
- Если ячейка $A[L]$ непуста, 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 $Insert$. ($1 \leqslant N \leqslant 131\,072$, $1 \leqslant M \leqslant 131\,072$).

В следующей строке даны N целых чисел L_i, которые описывают операции $Insert$ ($1 \leqslant L_i \leqslant M$).

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

key.in	key.out
5 4 3 3 4 1 3	6 4 0 5 2 3 1

Задача G. Permutations [10 sec, 256 mb]

Consider a cyclic alphabet which consists of the first ten letters of the usual Latin alphabet. It is called cyclic because the next letter after 'a' is 'b', the next after 'b' is 'c' and so on. The last letter 'j' is followed by the first letter 'a'.

You are given an initial string S which contains only the letters of this cyclic alphabet. You should process queries of three kinds:

- Reverse substring of S from L to R, inclusive.
- For substring of S from L to R, inclusive, replace each character with the D-th next character in the cyclic alphabetical order.
- For substring of S from L to R, inclusive, return the number of distinct permutations of the characters of this substring modulo $10^9 + 7$.

Формат входных данных

The first line of input contains one integer N ($1 \leqslant N \leqslant 10^5$) — the length of the string S, followed by the string S on the second line. The string contains only lowercase Latin letters from 'a' to 'j', inclusive. The third line contains an integer M ($1 \leqslant M \leqslant 10^5$) — the number of queries. It is followed by M lines, each of which represents one of the following three types of queries:

- `-1 L R` ($1 \leqslant L \leqslant R \leqslant N$) — reverse substring from L to R.
- `0 L R D` ($1 \leqslant L \leqslant R \leqslant N$, $0 < D \leqslant N$) — replace each letter with the D-th next letter in the cyclic alphabetical order.
- `1 L R` ($1 \leqslant L \leqslant R \leqslant N$) — return the number of distinct permutations of the characters of substring $[L, R]$ of S.

Формат выходных данных

For each query of type "`1 L R`", return the answer modulo $10^9 + 7$.

Пример

permutations.in	permutations.out
6 abcabc 3 -1 1 6 0 1 3 1 1 1 6	180

Задача H. Persistent List [3 sec, 256 mb]

Даны N списков. Каждый состоит из одного элемента.

Нужно научитсья совершать следующие операции:

- `merge` — взять два каких-то уже существующих списка и породить новый, равный их конкатенации.
- `head` — взять какой-то уже существующий список L и породить два новых, в одном первый элемент L, во втором весь L кроме первого элемента.
- `tail` — взять какой-то уже существующий список L и породить два новых, в одном весь L кроме последнего элемента, во втором последний элемент L.

Для свежесозданных списков нужно говорить сумму элементов в них по модулю $10^9 + 7$.

Формат входных данных

Число N ($1 \leqslant N \leqslant 10^5$). Далее N целых чисел от 1 до 10^9 — элементы списков. Исходные списки имеют номера — $1, 2, \cdots, N$.

Затем число M ($1 \leqslant M \leqslant 10^5$) — количество операций. Далее даны операции в следующем формате:

- `merge i j`
- `head i`
- `tail i`

Где `i` и `j` — номера уже существующих списков. Если в текущий момент имеется K списков, новый список получает номер $K + 1$.

Для операций `head` и `tail` считается, что сперва порождается левая часть, затем правая (см. пример). Также вам гарантируется, что никогда не будут порождаться пустые списки.

Формат выходных данных

Для каждого нового списка нужно вывести сумму элементов по модулю $10^9 + 7$.

Пример

plist.in	plist.out
4 1 2 3 4 7 merge 1 2 merge 3 4 merge 6 5 head 7 tail 9 merge 2 3 merge 1 1	3 7 10 3 7 5 2 5 2