

Содержание

Как решать задачи?	3
Обязательные задачи	4
Задача 16А. И снова сумма... [1 sec, 256 mb]	4
Задача 16В. К-ый максимум [1 sec, 256 mb]	5
Задача 16С. Неявный Ключ [1 sec, 256 mb]	6
Задача 16D. Range Minimum Query [1 sec, 256 mb]	7
Дополнительные задачи	8
Задача 16Е. Вперёд! [1 sec, 256 mb]	8
Задача 16F. Вставка ключевых значений [1 sec, 256 mb]	9
Задача 16G. Перестановки [1 sec, 256 mb]	10
Задача 16Н. Эх, дороги... [1 sec, 256 mb]	11

Пример работы с файлами.

Если вы не умеете читать/выводить данные, или открывать файлы, воспользуйтесь примерами. <http://acm.math.spbu.ru/~sk1/algo/sum/>

Пример работы с файлами.

В некоторых задачах большой ввод и вывод. Про ввод-вывод в C++:

http://acm.math.spbu.ru/~sk1/algo/input-output/cpp_common.html

Имеет смысл пользоваться супер быстрым вводом-выводом. Две версии:

http://acm.math.spbu.ru/~sk1/algo/input-output/io_export.cpp.html

http://acm.math.spbu.ru/~sk1/algo/input-output/fread_write_export.cpp.html

Выделение памяти.

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

<http://acm.math.spbu.ru/~sk1/algo/memory.cpp.html>

Как решать задачи?

Всё, что вам нужно написать, – какое-нибудь сбалансированное BST.

С лекций вы знаете два – AVL и Tgear.

Для лучшего усвоения материала реализуйте хотя бы в одной задаче AVL (то есть, можно без удаления).

Обязательные задачи

Задача 16А. И снова сумма... [1 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 он там уже есть, то множество не меняется);
- $sum(l, r)$ — 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 $sum(l, r)$.

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

Замечание

У вас уже была эта задача на тему “пополняемый массив”. В этот раз обязательно сдать её, используя BST.

Задача 16В. К-ый максимум [1 сек, 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

kthmax.in	kthmax.out
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 16С. Неявный Ключ [1 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

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

Задача 16D. Range Minimum Query [1 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

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

Дополнительные задачи

Задача 16Е. 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 ($2 \leq n \leq 100\,000$, $1 \leq m \leq 100\,000$) — число солдат и число приказов. Следующие m строк содержат приказы в виде двух целых чисел: l_i и r_i ($1 \leq l_i \leq r_i \leq n$).

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	1 4 5 2 3 6
2 4	
3 5	
2 2	

Задача 16F. Вставка ключевых значений [1 sec, 256 mb]

Вас наняла на работу компания MascoHard, 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, \dots, 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 $Insert$. ($1 \leq N \leq 131\,072$, $1 \leq M \leq 131\,072$).

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

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 — $A[1], A[2], \dots, A[W]$. Для пустых ячеек выводите нули.

Пример

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

Задача 16G. Перестановки [1 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:

- $-1 \ L \ R$ ($1 \leq L \leq R \leq N$) — запрос на разворот подстроки.
- $0 \ L \ R \ D$ ($1 \leq L \leq R \leq N, 0 < D \leq N$) — запрос на замену символов в подстроке.
- $1 \ L \ R$ ($1 \leq L \leq R \leq N$) — запрос на 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

permutations.in	permutations.out
6	180
abcsabc	
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.

Задача 16Н. Эх, дороги... [1 сек, 256 mb]

В многострадальном Тридесятом государстве опять готовится дорожная реформа. Впрочем, надо признать, дороги в этом государстве находятся в довольно плачевном состоянии. Так что реформа не повредит. Одна проблема — дорожникам не развернуться, поскольку в стране действует жесткий закон — из каждого города должно вести не более двух дорог. Все дороги в государстве двусторонние, то есть по ним разрешено движение в обоих направлениях (разумеется, разметка отсутствует). В результате реформы некоторые дороги будут строиться, а некоторые другие закрываться на бессрочный ремонт.

Петя работает диспетчером в службе грузоперевозок на дальние расстояния. В связи с предстоящими реформами, ему необходимо оперативно определять оптимальные маршруты между городами в условиях постоянно меняющейся дорожной ситуации. В силу большого количества пробок и сотрудников дорожной полиции в городах, критерием оптимальности маршрута считается количество промежуточных городов, которые необходимо проехать.

Помогите Пете по заданной последовательности сообщений об изменении структуры дорог и запросам об оптимальном способе проезда из одного города в другой, оперативно отвечать на запросы.

Формат входных данных

В первой строке входного файла заданы числа n — количество городов, m — количество дорог в начале реформы и q — количество сообщений об изменении дорожной структуры и запросов ($1 \leq n, m \leq 100\,000$, $q \leq 200\,000$). Следующие m строк содержат по два целых числа каждая — пары городов, соединенных дорогами перед реформой. Следующие q строк содержат по три элемента, разделенных пробелами. «+ i j » означает строительство дороги от города i до города j , «- i j » означает закрытие дороги от города i до города j , «? i j » означает запрос об оптимальном пути между городами i и j .

Гарантируется, что в начале и после каждого изменения никакие два города не соединены более чем одной дорогой, и из каждого города выходит не более двух дорог. Никакой город не соединяется дорогой сам с собой.

Формат выходных данных

На каждый запрос вида «? i j » выведите одно число — минимальное количество промежуточных городов на маршруте из города i в город j . Если проехать из i в j невозможно, выведите -1 .

Пример

roads.in	roads.out
5 4 6	0
1 2	-1
2 3	1
1 3	2
4 5	
? 1 2	
? 1 5	
- 2 3	
? 2 3	
+ 2 4	
? 1 5	