

Задача А. RMQ

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Реализуйте структуру данных, которая на данном массиве из N целых чисел позволяет узнать максимальное значение на этом массиве и индекс элемента, на котором достигается это максимальное значение.

Формат входных данных

В первой строке вводится натуральное число N ($1 \leq N \leq 10^5$) – количество элементов в массиве. В следующей строке содержатся N целых чисел, не превосходящих по модулю 10^9 – элементы массива. Гарантируется, что в массиве нет одинаковых элементов. Далее идет число K ($0 \leq K \leq 10^5$) – количество запросов к структуре данных. Каждая из следующих K строк содержит два целых числа l и r ($1 \leq l \leq r \leq N$) – левую и правую границы отрезка в массиве для данного запроса.

Формат выходных данных

Для каждого из запросов выведите два числа: наибольшее значение среди элементов массива на отрезке от l до r и индекс одного из элементов массива, принадлежащий отрезку от l до r , на котором достигается этот максимум.

Пример

стандартный ввод	стандартный вывод
5	7 1
7 3 1 6 4	6 4
3	1 3
1 5	
2 4	
3 3	

Задача В. Катый ноль

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

Реализуйте эффективную структуру данных, позволяющую изменять элементы массива и вычислять индекс k -го слева нуля на данном отрезке в массиве.

Формат входных данных

В первой строке вводится одно натуральное число N ($1 \leq N \leq 200\,000$) — количество чисел в массиве. Во второй строке вводятся N чисел от 0 до 100 000 — элементы массива. В третьей строке вводится одно натуральное число M ($1 \leq M \leq 200\,000$) — количество запросов. Каждая из следующих M строк представляет собой описание запроса. Сначала вводится одна буква, кодирующая вид запроса (**s** — вычислить индекс k -го нуля, **u** — обновить значение элемента). Следом за **s** вводится три числа — левый и правый концы отрезка и число k ($1 \leq k \leq N$). Следом за **u** вводятся два числа — номер элемента и его новое значение.

Формат выходных данных

Для каждого запроса s выведите результат. Все числа выводите в одну строку через пробел. Если нужного числа нулей на запрашиваемом отрезке нет, выводите -1 для данного запроса.

Пример

стандартный ввод	стандартный вывод
5 0 0 3 0 2 3 u 1 5 u 1 0 s 1 5 3	4

Задача С. Перестановки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.75 секунд
Ограничение по памяти: 512 мегабайт

Вася выписал на доске в каком-то порядке все числа от 1 по N , каждое число ровно по одному разу. Количество чисел оказалось довольно большим, поэтому Вася не может окинуть взглядом все числа. Однако ему надо всё-таки представлять эту последовательность, поэтому он написал программу, которая отвечает на вопрос — сколько среди чисел, стоящих на позициях с x по y , по величине лежат в интервале от k до l . Сделайте то же самое.

Формат входных данных

В первой строке лежит два натуральных числа — $1 \leq N \leq 10^5$ — количество чисел, которые выписал Вася и $1 \leq M \leq 10^5$ — количество вопросов, которые Вася хочет задать программе. Во второй строке дано N чисел — последовательность чисел, выписанных Васей. Далее в M строках находятся описания вопросов. Каждая строка содержит четыре целых числа $1 \leq x \leq y \leq N$ и $1 \leq k \leq l \leq N$.

Формат выходных данных

Выведите M строк, каждая должна содержать единственное число — ответ на Васин вопрос.

Пример

стандартный ввод	стандартный вывод
4 2	1
1 2 3 4	3
1 2 2 3	
1 3 1 3	

Замечание

Напиши merge-sort-tree. Другие решения будут забанены.

Задача D. Ближайшее большее число справа

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дан массив a из n чисел. Нужно обрабатывать запросы:

0. `set(i, x)` – присвоить новое значение элементу массива $a[i] = x$;
1. `get(i, x)` – найти $\min k: k \geq i$ и $a_k \geq x$.

Формат входных данных

Первая строка входных данных содержит два числа: длину массива n и количество запросов m ($1 \leq n, m \leq 200\,000$).

Во второй строке записаны n целых чисел – элементы массива a ($0 \leq a_i \leq 200\,000$).

Следующие m строк содержат запросы, каждый запрос содержит три числа t, i, x . Первое число t равно 0 или 1 – тип запроса. $t = 0$ означает запрос типа `set`, $t = 1$ соответствует запросу типа `get`, $1 \leq i \leq n$, $0 \leq x \leq 200\,000$. Элементы массива нумеруются с единицы.

Формат выходных данных

На каждый запрос типа `get` на отдельной строке выведите соответствующее значение k . Если такого k не существует, выведите -1 .

Пример

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

Задача Е. Дерево отрезков с операцией на отрезке

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Реализуйте эффективную структуру данных для хранения элементов и увеличения нескольких подряд идущих элементов на одно и то же число.

Формат входных данных

В первой строке вводится одно натуральное число N ($1 \leq N \leq 100\,000$) — количество чисел в массиве.

Во второй строке вводятся N чисел от 0 до 100 000 — элементы массива.

В третьей строке вводится одно натуральное число M ($1 \leq M \leq 30\,000$) — количество запросов.

Каждая из следующих M строк представляет собой описание запроса. Сначала вводится одна буква, кодирующая вид запроса (g — получить текущее значение элемента по его номеру, a — увеличить все элементы на отрезке).

Следом за g вводится одно число — номер элемента.

Следом за a вводится три числа — левый и правый концы отрезка и число add , на которое нужно увеличить все элементы данного отрезка массива ($0 \leq add \leq 100\,000$).

Формат выходных данных

Выведите в одну строку через пробел ответы на каждый запрос g .

Пример

стандартный ввод	стандартный вывод
5	4
2 4 3 5 2	2
5	14
g 2	5
g 5	
a 1 3 10	
g 2	
g 4	

Задача F. Число возрастающих подпоследовательностей

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Задана последовательность из n чисел a_1, a_2, \dots, a_n . Подпоследовательностью длины k этой последовательности называется набор индексов i_1, i_2, \dots, i_k , удовлетворяющий неравенствам $1 \leq i_1 < i_2 < \dots < i_k \leq n$. Подпоследовательность называется возрастающей, если выполняются неравенства $a_{i_1} < a_{i_2} < \dots < a_{i_k}$.

Необходимо найти число возрастающих подпоследовательностей наибольшей длины заданной последовательности a_1, \dots, a_n . Так как это число может быть достаточно большим, необходимо найти остаток от его деления на $10^9 + 7$.

Формат входных данных

Первая строка входного файла содержит целое число n ($1 \leq n \leq 10^5$). Вторая строка входного файла содержит n целых чисел: a_1, a_2, \dots, a_n . Все a_i не превосходят 10^9 по абсолютной величине.

Формат выходных данных

В выходной файл выведите ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
5 1 2 3 4 5	1
6 1 1 2 2 3 3	8

Задача G. [В-Х] Армия покемонов

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

Пикачу — милый и дружелюбный покемон, живущий в стае диких пикачу.

Однако недавно стало известно, что команда R хочет украсть всех этих покемонов! Тренер покемонов Андрей решил помочь Пикачу собрать армию для борьбы с командой R.

В первую очередь Андрей посчитал всех покемонов: их оказалось ровно n штук. Затем он установил силу каждого покемона и так получилось, что i -й покемон имеет силу, равную a_i .

В качестве армии Андрей может выбрать любую непустую подпоследовательность покемонов. Иными словами, Андрей выбирает какой-то массив b из k индексов таких, что $1 \leq b_1 < b_2 < \dots < b_k \leq n$, и его армия будет состоять из покемонов с силами $a_{b_1}, a_{b_2}, \dots, a_{b_k}$.

Сила армии вычисляется как знакопеременная сумма элементов подпоследовательности, то есть $a_{b_1} - a_{b_2} + a_{b_3} - a_{b_4} + \dots$.

Андрей экспериментирует с построением покемонов. Он q раз меняет двух покемонов местами, а именно, в i -й раз он менял местами покемонов с номерами l_i и r_i .

Андрею надо знать: какую максимальную силу армии он мог получить при начальной расстановке покемонов, а также после каждого изменения строя?

Помогите Андрею и покемонам, иначе команде R удастся воплотить в жизнь свой коварный план!

Формат входных данных

В первой строке каждого теста находятся два целых числа n и q ($1 \leq n \leq 3 \cdot 10^5, 0 \leq q \leq 3 \cdot 10^5$) — количество покемонов и количество обменов соответственно.

Во второй строке находятся n целых положительных чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$) — силы покемонов.

i -я из следующих q строк содержит два целых положительных числа l_i и r_i ($1 \leq l_i \leq r_i \leq n$) — номера обмениваемых покемонов в i -й операции.

Формат выходных данных

Выведите $q + 1$ число — максимально возможную силу армии до изменений и после каждого изменения.

Примеры

стандартный ввод	стандартный вывод
3 1 1 3 2 1 2	3 4
2 2 1 2 1 2 1 2	2 2 2
7 5 1 2 5 4 3 6 7 1 2 6 7 3 4 1 2 2 3	9 10 10 10 9 11

Задача Н. Разреженные таблицы

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

Дан массив из n чисел. Требуется написать программу, которая будет отвечать на запросы следующего вида: найти минимум на отрезке между u и v включительно.

Формат входных данных

В первой строке заданы три натуральных числа n , m ($1 \leq n \leq 10^5$, $1 \leq m \leq 10^7$) и a_1 ($0 \leq a_1 < 16\,714\,589$) — количество элементов в массиве, количество запросов и первый элемент массива соответственно. Вторая строка содержит два натуральных числа u_1 и v_1 ($1 \leq u_1, v_1 \leq n$) — первый запрос.

Для того, чтобы размер ввода был небольшой, массив и запросы генерируются.

Элементы a_2, a_3, \dots, a_n задаются следующей формулой:

$$a_{i+1} = (23 \cdot a_i + 21563) \bmod 16714589.$$

Например, при $n = 10$, $a_1 = 12345$ получается следующий массив: $a = (12345, 305498, 7048017, 11694653, 1565158, 2591019, 9471233, 570265, 13137658, 1325095)$.

Запросы генерируются следующим образом:

$$\begin{aligned} u_{i+1} &= ((17 \cdot u_i + 751 + r_i + 2i) \bmod n) + 1, \\ v_{i+1} &= ((13 \cdot v_i + 593 + r_i + 5i) \bmod n) + 1, \end{aligned}$$

где r_i — ответ на запрос номер i .

Обратите внимание, что u_i может быть больше, чем v_i .

Формат выходных данных

В выходной файл выведите u_m , v_m и r_m (последний запрос и ответ на него).

Примеры

стандартный ввод	стандартный вывод
10 8 12345 3 9	5 3 1565158

Замечание

Можно заметить, что массивы u , v и r можно не сохранять в памяти полностью.

Запросы и ответы на них выглядят следующим образом:

i	u_i	v_i	r_i
1	3	9	570265
2	10	1	12345
3	1	2	12345
4	10	10	1325095
5	5	9	570265
6	2	1	12345
7	3	2	305498
8	5	3	1565158

Эта задача скорее всего не решается стандартными интерпретаторами Python 2 и Python 3. Используйте соответствующие компиляторы PyPy.

Задача I. Прямоугольники

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

Когда-то тут была легенда про вёдра, но её съели.

Жюри олимпиады

Есть таблица T размера $N \times M$. Элементами таблицы являются прямоугольники T_{ij} , где $0 \leq i < N$ и $0 \leq j < M$. Прямоугольник T_{ij} задаётся четвёркой чисел $(x_1^{ij}, y_1^{ij}, x_2^{ij}, y_2^{ij})$, где (x_1^{ij}, y_1^{ij}) и (x_2^{ij}, y_2^{ij}) — координаты противоположных углов прямоугольника. Стороны прямоугольника параллельны осям координат.

Далее вам поступают запросы. Каждый запрос состоит из четырёх чисел: (r_1, c_1, r_2, c_2) . Ответом на такой запрос является площадь фигуры, являющейся пересечением всех прямоугольников T_{ij} таких, что $\min(r_1, r_2) \leq i \leq \max(r_1, r_2)$ и $\min(c_1, c_2) \leq j \leq \max(c_1, c_2)$. Запросов очень много, поэтому мы просим вас вывести сумму ответов на все запросы по модулю $10^9 + 7$.

Формат входных данных

В первой строке записаны два целых числа N и M — размеры таблицы T ($1 \leq N, M \leq 127$). Далее в N строках описывается таблица T : в $(i+1)$ -й строке $(j+1)$ -я четвёрка чисел $x_1^{ij} y_1^{ij} x_2^{ij} y_2^{ij}$ описывает прямоугольник T_{ij} . Гарантируется, что $|x_k^{ij}|, |y_k^{ij}| \leq 10^6$.

Дальше в отдельной строке записано четыре числа. Первое из них, число Q — количество запросов ($1 \leq Q \leq 5 \cdot 10^6$). Следующие три числа — это A, B, v_0 ($0 \leq A, B, v_0 < 10^9 + 7$). При помощи этих чисел генерируется бесконечная последовательность $\{v_i\}$ по правилу $v_i = (A \cdot v_{i-1} + B) \bmod (10^9 + 7)$.

После этого k -й запрос (запросы нумеруются с единицы) задаётся следующей четвёркой чисел: $(v_{4k-3} \bmod N, v_{4k-2} \bmod M, v_{4k-1} \bmod N, v_{4k} \bmod M)$.

Формат выходных данных

Выведите сумму ответов на все запросы по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
2 2 0 0 2 2 1 1 3 3 0 3 2 1 1 2 3 0 1 500000003 4 2	1
3 2 8 -1 -7 6 6 8 9 10 -4 -10 4 9 -3 -8 6 9 -2 -9 3 8 -5 7 7 3 5 303164476 273973578 65779139	85

Замечание

В первом примере запрос имеет вид $(1, 0, 0, 1)$, то есть это запрос ко всей таблице. Пересечением всех прямоугольников является квадрат с углами в точках $(1, 1)$ и $(2, 2)$. Его площадь равна 1.

Во втором примере запросы имеют вид $(0, 1, 1, 1)$, $(1, 0, 2, 0)$, $(0, 0, 2, 1)$, $(0, 1, 1, 1)$, $(0, 1, 0, 0)$. На второй запрос ответ — 85, на остальные — 0.