

5 - Запросы на отрезках

A. Сумма простая

1 секунда, 512 мегабайт

Вам нужно научиться отвечать на запрос «сумма чисел на отрезке».

Массив не меняется. Запросов много. Отвечать на каждый запрос следует за  $\mathcal{O}(1)$ .

Входные данные

Размер массива —  $n$  и числа  $x, y, a_0$ , порождающие массив  $a$ :  
 $a_i = (x \cdot a_{i-1} + y) \bmod 2^{16}$

Далее следует количество запросов  $m$  и числа  $z, t, b_0$ , порождающие массив  $b$ :  $b_i = (z \cdot b_{i-1} + t) \bmod 2^{30}$ .

Массив  $c$  строится следующим образом:  $c_i = b_i \bmod n$ .

Запросы:  $i$ -й из них — найти сумму на отрезке от  $\min(c_{2i}, c_{2i+1})$  до  $\max(c_{2i}, c_{2i+1})$  в массиве  $a$ .

Ограничения:  $1 \leq n \leq 10^7, 0 \leq m \leq 10^7$ . Все числа целые от 0 до  $2^{16}$ .  $t$  может быть равно  $-1$ .

Выходные данные

Выведите сумму всех сумм.

входные данные
3 1 2 3 3 1 -1 4
выходные данные
23

$a = \{3, 5, 7\}, b = \{4, 3, 2, 1, 0, 2^{30} - 1\}, c = \{1, 0, 2, 1, 0, 0\}$ ,  
запросы =  $\{[0, 1], [1, 2], [0, 0]\}$ , суммы =  $\{8, 12, 3\}$ .

B. RSQ

2 секунды, 256 мегабайт

Входные данные

В первой строке находится число  $n$  — размер массива.  
( $1 \leq n \leq 500\,000$ ) Во второй строке находится  $n$  чисел  $a_i$  — элементы массива. Далее содержится описание операций, их количество не превышает  $1\,000\,000$ . В каждой строке находится одна из следующих операций:

- $\text{set } i\ x$  — установить  $a[i]$  в  $x$ .
- $\text{sum } i\ j$  — вывести значение суммы элементов в массиве на отрезке с  $i$  по  $j$ , гарантируется, что  $(1 \leq i \leq j \leq n)$ .

Все числа во входном файле и результаты выполнения всех операций не превышают по модулю  $10^{18}$ .

Выходные данные

Выведите последовательно результат выполнения всех операций  $\text{sum}$ . Следуйте формату выходного файла из примера.

входные данные
5 1 2 3 4 5 sum 2 5 sum 1 5 sum 1 4 sum 2 4 set 1 10 set 2 3 set 5 2 sum 2 5 sum 1 5 sum 1 4 sum 2 4

выходные данные

14  
15  
10  
9  
12  
22  
20  
10

C. RMQ2

2 секунды, 256 мегабайт

Входные данные

В первой строке находится число  $n$  — размер массива. ( $1 \leq n \leq 10^5$ )  
Во второй строке находится  $n$  чисел  $a_i$  — элементы массива. Далее содержится описание операций, их количество не превышает  $2 \cdot 10^5$ .  
В каждой строке находится одна из следующих операций:

- $\text{set } i\ j\ x$  — установить все  $a[k], i \leq k \leq j$  в  $x$ .
- $\text{add } i\ j\ x$  — увеличить все  $a[k], i \leq k \leq j$  на  $x$ .
- $\text{min } i\ j$  — вывести значение минимального элемента в массиве на отрезке с  $i$  по  $j$ , гарантируется, что  $(1 \leq i \leq j \leq n)$ .

Все числа во входном файле и результаты выполнения всех операций не превышают по модулю  $10^{18}$ .

Выходные данные

Выведите последовательно результат выполнения всех операций  $\text{min}$ . Следуйте формату выходного файла из примера.

входные данные
5 1 2 3 4 5 min 2 5 min 1 5 min 1 4 min 2 4 set 1 3 10 add 2 4 4 min 2 5 min 1 5 min 1 4 min 2 4
выходные данные
2 1 1 2 5 5 8 8

D. Разреженные таблицы

2 секунды, 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.

Входные данные

В первой строке даны 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, 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:

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

где  $r_i$  — ответ на запрос номер  $i$ .  
Обратите внимание, что  $u_i$  может быть больше, чем  $v_i$ .

**Выходные данные**

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

входные данные
10 8 12345 3 9
выходные данные
5 3 1565158

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.

**Е. Криптография**

2 секунды, 256 мегабайт

Задано  $n$  матриц  $A_1, A_2, \dots, A_n$  размера  $2 \times 2$ . Необходимо для нескольких запросов вычислить произведение матриц  $A_i, A_{i+1}, \dots, A_j$ . Все вычисления производятся по модулю  $r$ .

**Входные данные**

Первая строка входного файла содержит числа  $r$  ( $1 \leq r \leq 10\,000$ ),  $n$  ( $1 \leq n \leq 200\,000$ ) и  $m$  ( $1 \leq m \leq 200\,000$ ). Следующие  $n$  блоков по две строки содержащие по два числа в строке — описания матриц. Затем следуют  $m$  пар целых чисел от 1 до  $n$ , запросы на произведение на отрезке.

**Выходные данные**

Выведите  $m$  блоков по две строки, по два числа в каждой — произведения на отрезках. Разделяйте блоки пустой строкой. Все вычисления производятся по модулю  $r$ .

входные данные
3 4 4 0 1 0 0  2 1 1 2  0 0 0 2  1 0 0 2  1 4 2 3 1 3 2 2
выходные данные
0 2 0 0  0 2 0 1  0 1 0 0  2 1 1 2

**F. RMQ наоборот**

2 секунды, 256 мегабайт

Рассмотрим массив  $a[1..n]$ . Пусть  $Q(i, j)$  — ответ на запрос о нахождении минимума среди чисел  $a[i], \dots, a[j]$ . Вам даны несколько запросов и ответы на них. Восстановите исходный массив.

**Входные данные**

Первая строка входного файла содержит число  $n$  — размер массива, и  $m$  — число запросов ( $1 \leq n, m \leq 100\,000$ ). Следующие  $m$  строк содержат по три целых числа  $i, j$  и  $q$ , означающих, что  $Q(i, j) = q$  ( $1 \leq i \leq j \leq n, -2^{31} \leq q \leq 2^{31} - 1$ ).

**Выходные данные**

Если искомого массива не существует, выведите строку «inconsistent».

В противном случае в первую строку выходного файла выведите «consistent». Во вторую строку выходного файла выведите элементы массива. Элементами массива должны быть целые числа в интервале от  $-2^{31}$  до  $2^{31} - 1$  включительно. Если решений несколько, выведите любое.

входные данные
3 2 1 2 1 2 3 2
выходные данные
consistent 1 2 2

входные данные
3 3 1 2 1 1 1 2 2 3 2
выходные данные
inconsistent

---

[Codeforces](#) (c) Copyright 2010-2020 Михаил Мирзаянов  
Соревнования по программированию 2.0