

Содержание

Разминка	2
1 Задача А. Сумма [2 секунд, 256 mb]	2
2 Задача В. Range Variation Query [2 секунд, 256 mb]	3
3 Задача С. Сумма [2 секунд, 256 mb]	4
4 Задача D. RMQ [1 секунд, 256 mb]	5
Задачи	6
5 Задача Е. Знакопереживание [2 секунд, 256 mb]	6
6 Задача F. Прямоугольники [2 секунд, 256 mb]	7
7 Задача G. K -инверсии [2 секунд, 256 mb]	8
8 Задача H. RMQ. RMQ Наоборот (*) [3 секунд, 256 mb]	9
9 Задача I. Наибольшая общая возрастающая [500 секунд, 256 mb]	10

Разминка

1 Задача А. Сумма [2 секунд, 256 mb]

Имя входного файла: `sum.in`
Имя выходного файла: `sum.out`

Дан массив из N элементов, нужно научиться находить сумму чисел на отрезке.

Формат входных данных

Первая строка входного файла содержит два целых числа N и K — число чисел в массиве и количество запросов. ($1 \leq N \leq 100\,000$), ($0 \leq K \leq 100\,000$). Следующие K строк содержат запросы

1. $A\ i\ x$ — присвоить i -му элементу массива значение x ($1 \leq i \leq n$, $0 \leq x \leq 10^9$)
2. $Q\ l\ r$ — найти сумму чисел в массиве на позициях от l до r . ($1 \leq l \leq r \leq n$)

Изначально в массиве живут нули.

Формат выходных данных

На каждый запрос вида $Q\ l\ r$ нужно вывести единственное число — сумму на отрезке.

Примеры

sum.in	sum.out
5 9	0
A 2 2	2
A 3 1	1
A 4 2	2
Q 1 1	0
Q 2 2	5
Q 3 3	
Q 4 4	
Q 5 5	
Q 1 5	

2 Задача В. Range Variation Query [2 секунд, 256 mb]

Имя входного файла: `rvq.in`

Имя выходного файла: `rvq.out`

В начальный момент времени последовательность a_n задана следующей формулой:
 $a_n = n^2 \bmod 12345 + n^3 \bmod 23456$.

Требуется много раз отвечать на запросы следующего вида:

- найти разность между максимальным и минимальным значениями среди элементов a_i, a_{i+1}, \dots, a_j ;
- присвоить элементу a_i значение j .

Формат входных данных

Первая строка входного файла содержит натуральное число k — количество запросов ($1 \leq k \leq 100\,000$). Следующие k строк содержат запросы, по одному на строке. Запрос номер i описывается двумя целыми числами x_i, y_i .

Если $x_i > 0$, то требуется найти разность между максимальным и минимальным значениями среди элементов a_{x_i}, \dots, a_{y_i} . При этом $1 \leq x_i \leq y_i \leq 100\,000$.

Если $x_i < 0$, то требуется присвоить элементу $a_{|x_i|}$ значение y_i . В этом случае $-100\,000 \leq x_i \leq -1$ и $|y_i| \leq 100\,000$.

Формат выходных данных

Для каждого запроса первого типа в выходной файл требуется вывести одну строку, содержащую разность между максимальным и минимальным значениями на соответствующем отрезке.

Примеры

rvq.in	rvq.out
7	34
1 3	68
2 4	250
-2 -100	234
1 5	1
8 9	
-3 -101	
2 3	

3 Задача С. Сумма [2 секунд, 256 mb]

Имя входного файла: `sum.in`
Имя выходного файла: `sum.out`

Дан массив из N элементов, нужно научиться находить сумму чисел на отрезке.

Формат входных данных

Первая строка входного файла содержит два целых числа N и K — число чисел в массиве и количество запросов. ($1 \leq N \leq 100\,000$), ($0 \leq K \leq 100\,000$). Следующие K строк содержат запросы:

1. `A l r x` — присвоить элементам массива с позициями от l до r значение x ($1 \leq l \leq r \leq N$, $0 \leq x \leq 10^9$)
2. `Q l r` — найти сумму чисел в массиве на позициях от l до r . ($1 \leq l \leq r \leq N$)

Изначально массив заполнен нулями.

Формат выходных данных

На каждый запрос вида `Q l r` нужно вывести единственное число — сумму на отрезке.

Примеры

sum.in	sum.out
5 9	3
A 2 3 2	2
A 3 5 1	3
A 4 5 2	4
Q 1 3	2
Q 2 2	7
Q 3 4	
Q 4 5	
Q 5 5	
Q 1 5	

4 Задача D. RMQ [1 секунд, 256 mb]

Имя входного файла: `stdin`

Имя выходного файла: `stdout`

Дан массив $a[1..n]$. Требуется написать программу, обрабатывающую два типа запросов.

- Запрос “ $\text{max } l \ r$ ”. Требуется найти максимум в массиве a от l -ой ячейки до r -ой включительно.
- Запрос “ $\text{add } l \ r \ v$ ”. Требуется прибавить значение v к каждой ячейке массива a от l -ой до r -ой включительно.

Формат входных данных

Первая строка содержит два целых числа n и q ($1 \leq n, q \leq 10^5$) – длина массива и число запросов соответственно. Вторая строка содержит n целых чисел a_1, \dots, a_n ($|a_i| \leq 10^5$), задающих соответствующие значения массива. Следующие q строк содержат запросы.

В зависимости от типа запрос может иметь вид либо “ $\text{max } l \ r$ ”, либо “ $\text{add } l \ r \ v$ ”. При этом $1 \leq l \leq r \leq 10^5$, $|v| \leq 10^5$.

Формат выходных данных

Для каждого запроса вида “ $\text{max } l \ r$ ” требуется в отдельной строке выдать значение соответствующего максимума.

Примеры

stdin	stdout
5 3	3
1 2 3 4 -5	7
max 1 3	
add 1 2 5	
max 1 3	

Задачи

5 Задача Е. Знакопереживание [2 секунд, 256 mb]

Имя входного файла: `signchange.in`
Имя выходного файла: `signchange.out`

Реализуйте структуру данных из n элементов $a_1, a_2 \dots a_n$, поддерживающую следующие операции:

- присвоить элементу a_i значение j ;
- найти знакопереживающую сумму на отрезке от l до r включительно $(a_l - a_{l+1} + a_{l+2} - \dots \pm a_r)$.

Формат входных данных

В первой строке входного файла содержится натуральное число n ($1 \leq n \leq 10^5$) — длина массива. Во второй строке записаны начальные значения элементов (неотрицательные целые числа, не превосходящие 10^4).

В третьей строке находится натуральное число m ($1 \leq m \leq 10^5$) — количество операций. В последующих m строках записаны операции:

- операция первого типа задается тремя числами $0 \ i \ j$ ($1 \leq i \leq n, 1 \leq j \leq 10^4$).
- операция второго типа задается тремя числами $1 \ l \ r$ ($1 \leq l \leq r \leq n$).

Формат выходных данных

Для каждой операции второго типа выведите на отдельной строке соответствующую знакопереживающую сумму.

Пример

<code>signchange.in</code>	<code>signchange.out</code>
3	-1
1 2 3	2
5	-1
1 1 2	3
1 1 3	
1 2 3	
0 2 1	
1 1 3	

6 Задача F. Прямоугольники [2 секунд, 256 mb]

Имя входного файла: `rects.in`
Имя выходного файла: `rects.out`

На плоскости задано n прямоугольников, никакие два из которых не имеют общих точек. В каждом прямоугольнике записано целое число.

Скажем, что прямоугольник B лежит *дальше* прямоугольника A , если левый верхний угол прямоугольника B лежит строго ниже и правее правого нижнего угла прямоугольника A .

Последовательность прямоугольников R_1, R_2, \dots, R_k назовем *цепью*, если для всех i прямоугольник R_i лежит дальше прямоугольника R_{i-1} . *Весом* цепи назовем сумму чисел, записанных во входящих в нее прямоугольниках.

Требуется найти цепь прямоугольников с максимальным весом.

Формат входных данных

Первая строка входного файла содержит число n — количество прямоугольников ($1 \leq n \leq 100\,000$).

Пусть ось x направлена слева направо, а ось y — снизу вверх. Следующие n строк содержат по пять целых чисел — координаты $x_{i,1}, y_{i,1}$ левого нижнего, $x_{i,2}, y_{i,2}$ правого верхнего углов прямоугольника и a_i — число, записанное в прямоугольнике. Координаты не превышают 10^9 по абсолютной величине. Числа, записанные в прямоугольниках, положительные и не превышают 10^9 . Ни один прямоугольник не лежит внутри другого.

Формат выходных данных

В первой строке выходного файла выведите одно число — максимальный возможный вес цепи прямоугольников. Во второй строке выведите через пробелы номера прямоугольников, образующих такую цепь, в порядке цепи. Если оптимальных решений несколько, разрешается вывести любое из них.

Пример

<code>rects.in</code>	<code>rects.out</code>
4	10
1 1 2 2 6	3 2
3 1 4 2 5	
0 3 1 4 5	
5 1 6 2 4	

7 Задача G. K -инверсии [2 секунд, 256 mb]

Имя входного файла: `kinverse.in`

Имя выходного файла: `kinverse.out`

Пусть дана перестановка a_1, a_2, \dots, a_n . Назовем k -инверсией набор чисел i_1, i_2, \dots, i_k таких, что $1 \leq i_1 < i_2 < \dots < i_k \leq n$ и $a_{i_1} > a_{i_2} > \dots > a_{i_k}$. Ваша задача — подсчитать количество различных k -инверсий в заданной перестановке.

Формат входных данных

В первой строке входного файла находятся число n — длина перестановки ($1 \leq n \leq 20\,000$), и число k ($2 \leq k \leq 10$). Во второй строке n чисел — сама перестановка.

Формат выходных данных

В выходной файл выведите единственное число — количество k -инверсий в заданной перестановке по модулю 10^9 .

Пример

kinverse.in	kinverse.out
3 2 3 1 2	2
5 3 5 4 3 2 1	10

8 Задача Н. RMQ. RMQ Наоборот (*) [3 секунд, 256 mb]

Имя входного файла: `rmq.in`
Имя выходного файла: `rmq.out`

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.

Формат входных данных

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 $Q(i, j) = q$ ($1 \leq i \leq j \leq n$, $-2^{31} \leq q \leq 2^{31} - 1$).

Формат выходных данных

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.

Примеры

<code>rmq.in</code>	<code>rmq.out</code>
3 2 1 2 1 2 3 2	<code>consistent</code> 1 2 3
3 3 1 2 1 1 1 2 2 3 2	<code>inconsistent</code>

9 Задача I. Наибольшая общая возрастающая [500 секунд, 256 mb]

Имя входного файла: `lcis.in`
Имя выходного файла: `lcis.out`

Даны две последовательности чисел — a и b . Нужно найти наибольшую общую возрастающую подпоследовательность. Более формально: такие $1 \leq i_1 < i_2 < \dots < i_k \leq a.n$ и $1 \leq j_1 < j_2 < \dots < j_k \leq b.n$, что $\forall t: a_{i_t} = b_{j_t}, a_{i_t} < a_{i_{t+1}}$ и k максимально.

Формат входных данных

На первой строке целые числа n и m от 1 до 3 000 — длины последовательностей. Вторая строка содержит n целых чисел, задающих первую последовательность. Третья строка содержит m целых чисел, задающих вторую последовательность. Все элементы последовательностей — целые неотрицательные числа, не превосходящие 10^9 .

Формат выходных данных

Выведите одно целое число — длину наибольшей общей возрастающей подпоследовательности.

Система оценки

Подзадача 1 (50 баллов) $n \leq 400$.
Подзадача 2 (50 баллов) $n \leq 3\,000$.

Пример

lcis.in	lcis.out
6 5 1 2 1 2 1 3 2 1 3 2 1	2