

## Содержание

<b>Задачи здорового человека</b>	<b>2</b>
<b>Задача 3А. Сумма [0.5 sec, 256 mb]</b>	<b>2</b>
<b>Задача 3В. Звёзды [0.2 sec, 256 mb]</b>	<b>3</b>
<b>Для искателей острых ощущений</b>	<b>4</b>
<b>Задача 3С. RMQ [0.7 sec, 256 mb]</b>	<b>4</b>
<b>Задача 3D. Окна [1 sec, 256 mb]</b>	<b>5</b>

---

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

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

Обратите внимание на GNU C++ компиляторы с суффиксом inc, они позволяют пользоваться **дополнительной библиотекой**. Под ними можно сдать **вот это**.

## Задачи здорового человека

### Задача 3А. Сумма [0.5 sec, 256 mb]

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

#### Формат входных данных

Первая строка 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

- “A  $i$   $x$ ” — присвоить  $i$ -му элементу массива значение  $x$  ( $1 \leq i \leq n$ ,  $0 \leq x \leq 10^9$ )
- “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

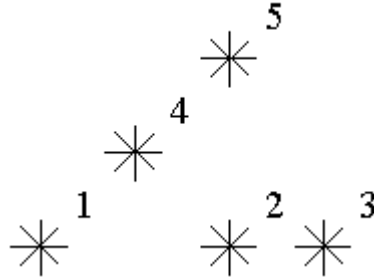
stdin	stdout
5 9	0
A 2 2	2
A 3 1	1
A 4 2	2
Q 1 1	0
Q 2 2	5
Q 3 3	
Q 4 4	
Q 5 5	
Q 1 5	

#### Замечание

Обыкновенное дерево отрезков. Оно же пригодится в следующей задаче ;)   
BST (AVL, treap) не должны заходить по времени.

### Задача 3В. Звёзды [0.2 sec, 256 mb]

Астрономы часто исследуют звёздные карты, на которых звёзды представлены точками на плоскости, каждая звезда имеет декартовы координаты. Пусть уровень звезды – количество звёзд, которые не выше и не правее данной звезды. Астрономы хотят найти распределение уровней звёзд.



Для примера посмотрим на карту звёзд на картинке выше. Уровень звезды номер 5 равен 3 (т.к. есть звёзды с номерами 1, 2, 4). Уровни звёзд 2 и 4 равны 1. На данной карте есть только одна звезда на уровне 0, две звезды на уровне 1, одна звезда на уровне 2 и одна звезда на уровне 3. Напишите программу, считающую количество звёзд на каждом уровне.

#### Формат входных данных

Вам дан один или несколько тестов. Каждый тест описывается следующим образом.

В первой строке количество звёзд  $N$  ( $1 \leq N \leq 15\,000$ ). Следующие  $N$  строк описывают координаты звёзд (два целых числа  $X$  и  $Y$ , разделённые пробелом,  $0 \leq X, Y \leq 32\,000$ ). В каждой точке плоскости находится не более одной звезды. Звёзды перечислены в порядке возрастания  $Y$  координаты, при равенстве в порядке возрастания  $X$  координаты.

#### Формат выходных данных

Выведите ответ для каждого теста. Ответ для теста описывается следующим образом.  $N$  строк, по одному числу в строке.  $i$ -я строка содержит количество звёзд на уровне  $i$  ( $i = 0 \dots N-1$ ).

#### Примеры

stdin	stdout
5	1
1 1	2
5 1	1
7 1	1
3 3	0
5 5	1
5	2
1 1	1
5 1	1
7 1	0
3 3	
5 5	

#### Замечание

Простейший scanline. Разобран на практике.

Подходящее дерево отрезков написано в предыдущей задаче.

## Для искателей острых ощущений

### Задача 3С. RMQ [0.7 sec, 256 mb]

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

- “max  $l r$ ”. Найти максимум в массиве  $a$  от  $l$ -ой ячейки до  $r$ -ой включительно.
- “add  $l r v$ ”. Прибавить значение  $v$  к каждой ячейке  $a[i]$  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 “max  $l r$ ”, либо “add  $l r v$ ”.

$1 \leq l \leq r \leq n, |v| \leq 10^5$ .

#### Формат выходных данных

Для каждого запроса вида “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	

#### Замечание

Дерево отрезков с отложенными операциями.

### Задача 3D. Окна [1 sec, 256 mb]

На экране расположены прямоугольные окна, каким-то образом перекрывающиеся (со сторонами, параллельными осям координат). Вам необходимо найти точку, которая покрыта наибольшим числом из них.

#### Формат входных данных

В первой строке входного файла записано число окон  $n$  ( $1 \leq n \leq 50\,000$ ).

Следующие  $n$  строк содержат координаты окон  $x_{(1,i)} y_{(1,i)} x_{(2,i)} y_{(2,i)}$ , где  $\langle x_{(1,i)}, y_{(1,i)} \rangle$  — координаты левого верхнего угла  $i$ -го окна, а  $\langle x_{(2,i)}, y_{(2,i)} \rangle$  — правого нижнего (на экране компьютера  $y$  растёт сверху вниз, а  $x$  — слева направо).

Все координаты — целые числа, по модулю не превосходящие  $2 \cdot 10^5$ .

#### Формат выходных данных

В первой строке выходного файла выведите максимальное число окон, покрывающих какую-либо из точек в данной конфигурации. Во второй строке выведите два целых числа, разделенные пробелом — координаты точки, покрытой максимальным числом окон. Окна считаются замкнутыми, т.е. покрывающими свои граничные точки.

#### Пример

stdin	stdout
2	2
0 0 3 3	3 2
1 1 4 4	

#### Замечание

Scanline. Похожая задача разобрана на практике.