

Содержание

Must have	2
Задача 28A. Сумма [0.1 sec, 256 mb]	2
Задача 28B. Звёзды [0.1 sec, 256 mb]	3
Задачи здорового человека	4
Задача 28C. RMQ [0.4 sec, 256 mb]	4
Задача 28D. Художник [0.6 sec, 256 mb]	5
Задача 28E. Окна [0.5 sec, 256 mb]	6
Задача 28F. Прямоугольники [0.5 sec, 256 mb]	7
Для искателей острых ощущений	8
Задача 28G. k-я статистика на отрезке [4.5 sec, 256 mb]	8
Задача 28H. Различные числа [0.5 sec, 256 mb]	10
Задача 28I. Треугольник [4.5 sec, 256 mb]	11
Для мастеров AI	12
Задача 28J. Фыр-Фыр-Фенвик [0.55 sec, 128 mb]	13

У вас не получается читать/выводить данные?

Воспользуйтесь примерами (**c++**) (**python**).

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

Обратите внимание на GNU C++ компиляторы с суффиксом inc.

Подни можно пользоваться **дополнительной библиотекой** (optimization.h).

То есть, использовать быстрый ввод-вывод: **пример про числа и строки**.

И быструю аллокацию памяти (ускоряет vector-set-map-весь-STL): **пример**.

Для тех, кто хочет разобраться, как всё это работает.

Короткая версия быстрого ввода-вывода (**тык**) и короткая версия аллокатора (**тык**).

Must have

Задача 28А. Сумма [0.1 сек, 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	

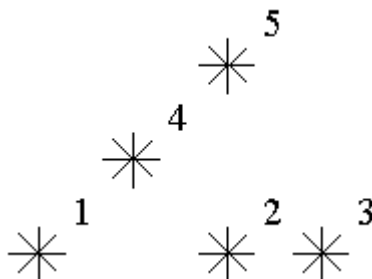
Замечание

Обыкновенное дерево отрезков.

Попробуйте написать «снизу», получится супер короткий и простой код.

Задача 28В. Звёзды [0.1 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 ($1 \leq N \leq 300\,000$). Следующие N строк описывают координаты звёзд (два целых числа X и Y , разделённые пробелом, $0 \leq X, Y < 10^6$). В каждой точке плоскости находится не более одной звезды. Звёзды перечислены в порядке возрастания Y координаты, при равенстве в порядке возрастания X координаты.

Сумма N по всем тестам также не больше 300 000.

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.

Задачи здорового человека

Задача 28С. RMQ [0.4 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 $a[]$ 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 28D. Художник [0.6 sec, 256 mb]

Итальянский художник-абстракционист Ф. Мандарино увлекся рисованием одномерных черно-белых картин. Он пытается найти оптимальное местоположение и количество черных участков картины. Для этого он проводит на прямой белые и черные отрезки, и после каждой из таких операций хочет знать количество черных отрезков на получившейся картине и их суммарную длину.

Изначально прямая — белая. Ваша задача — написать программу, которая после каждой из таких операций выводит в выходной файл интересные художника данные.

Формат входных данных

В первой строке входного файла содержится общее количество нарисованных отрезков ($1 \leq N \leq 100\,000$). В последующих N строках содержится описание операций. Каждая операция описывается строкой вида $c\ x\ l$, где c — цвет отрезка (W для белых отрезков, B для черных), а сам отрезок имеет вид $[x; x + l)$, причем координаты обоих концов — целые числа, не превосходящие по модулю 500 000. Длина задается положительным целым числом.

Формат выходных данных

После выполнения каждой из операций необходимо вывести в выходной файл на отдельной строке количество черных отрезков на картине и их суммарную длину, разделенные одним пробелом.

Пример

stdin	stdout
7	0 0
W 2 3	1 2
B 2 2	1 4
B 4 2	1 4
B 3 2	2 6
B 7 2	3 5
W 3 1	0 0
W 0 10	

Замечание

Минизачёт по теме «дерево отрезков». Разобрана на практике.

Задача 28Е. Окна [0.5 сек, 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.

Задача 28F. Прямоугольники [0.5 sec, 256 mb]

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

stdin	stdout
4	10
1 1 2 2 6	3 2
3 1 4 2 5	
0 3 1 4 5	
5 1 6 2 4	

Замечание

Продолжаем тренировать scanline. Предполагается, что это решение получается копированием уже сданной E и уже написанного дерева отрезков с нужными операциями.

Для искателей острых ощущений

Задача 28G. k -я статистика на отрезке [4.5 сек, 256 mb]

k -ю статистику на отрезке $[l..r]$ массива A можно определить следующим способом:

```
int get( int l, int r, int k ) {  
    B = отрезок [l..r] массива A  
    sort(B)  
    return B[k]  
}
```

Дан массив. Ваша задача — много раз отвечать на запрос “ k -я статистика на отрезке”.

Формат входных данных

Первая строка содержит целое число N , количество чисел в массиве ($1 \leq N \leq 450\,000$).

Вторая строка используется, чтобы сгенерировать массив $a_1, a_2, \dots, a_N$.

Она содержит три целых числа a_1, l и m ($0 \leq a_1, l, m < 10^9$).

$$a_i = (a_{i-1} \cdot l + m) \bmod 10^9, \quad 2 \leq i \leq N$$

Третья строка содержит целое число B — число групп запросов ($1 \leq B \leq 1000$). В следующих строках описывают группы запросов. Каждая группа описывается десятью целыми числами. Первым идет число G — количество запросов. Затем следуют x_1, l_x и m_x , потом y_1, l_y и m_y , и наконец, k_1, l_k и m_k ($1 \leq x_1 \leq y_1 \leq N$, $1 \leq k_1 \leq y_1 - x_1 + 1$, $0 \leq l_x, m_x, l_y, m_y, l_k, m_k < 10^9$). Они используются, чтобы сгенерировать вспомогательную последовательность x_g и y_g и текущие параметры i_g, j_g и k_g для $1 \leq g \leq G$:

$$\begin{aligned} x_g &= ((i_{g-1} - 1) \cdot l_x + m_x) \bmod N + 1, & 2 \leq g \leq G \\ y_g &= ((j_{g-1} - 1) \cdot l_y + m_y) \bmod N + 1, & 2 \leq g \leq G \\ i_g &= \min(x_g, y_g), & 1 \leq g \leq G \\ j_g &= \max(x_g, y_g), & 1 \leq g \leq G \\ k_g &= (((k_{g-1} - 1) \cdot l_k + m_k) \bmod (j_g - i_g + 1)) + 1, & 2 \leq g \leq G \end{aligned}$$

Сгенерированные параметры означают, что в g -м запросе, Нужно узнать k_g -ую статистику на отрезке $[i_g, j_g]$ массива. Общее количество запросов по всем группам не превышает 600 000.

Формат столь необычный, чтобы тесты были маленькие по объёму.

Формат выходных данных

Выведите одно число: сумму всех полученных статистик.

Пример

stdin	stdout
5 1 1 1 5 1 1 0 0 3 0 0 2 0 0 1 2 0 0 5 0 0 3 0 0 1 1 0 0 5 0 0 5 0 0 1 3 0 0 3 0 0 1 0 0 1 1 0 0 4 0 0 1 0 0	15

Замечание

Будьте аккуратны при генерации запросов. Часто ошибаются именно в этой части.

Подсказка по решению

Задача имеет простое решение персистентным деревом отрезков за $\mathcal{O}(\log n)$ на запрос.

Задача 28Н. Различные числа [0.5 sec, 256 mb]

Сколько различных чисел на отрезке массива?

Формат входных данных

На первой строке длина массива n ($1 \leq n \leq 300\,000$). На второй строке n целых чисел от 0 до 10^9-1 . На третьей строке количество запросов q ($1 \leq q \leq 300\,000$). Следующие q строк содержат описание запросов, по одному на строке. Каждый запрос задаётся парой целых чисел l, r ($1 \leq l \leq r \leq n$).

Формат выходных данных

Выведите ответы на запросы по одному в строке.

Примеры

stdin	stdout
5	3
1 1 2 1 3	2
3	3
1 5	
2 4	
3 5	

Задача 28I. Треугольник [4.5 сек, 256 mb]

Ваша задача — написать программу, хранящую мультимножество точек и позволяющую отвечать на запросы двух видов:

- добавить точку в множество,
- посчитать количество точек множества, лежащих внутри или на границе данного треугольника.

Формат входных данных

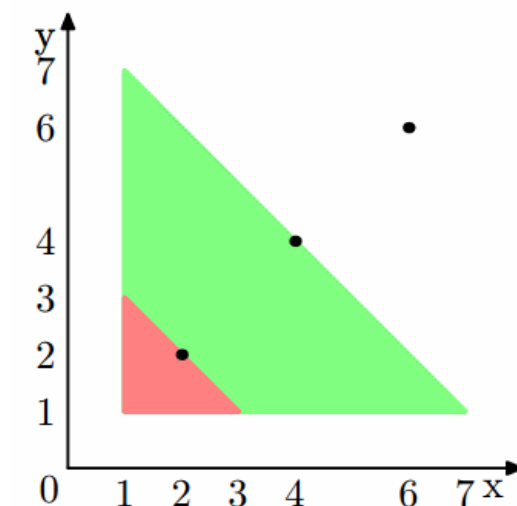
На первой строке число запросов m ($1 \leq m \leq 100\,000$). Следующие m строк содержат или $1 \ x \ y$ или $2 \ x \ y \ r$. Запрос 2-го типа представлен треугольником с углами в точках (x, y) , $(x + r, y)$, $(x, y + r)$. Известно, что $|x|, |y|, r \leq 10^8$ и $r > 0$.

Формат выходных данных

Для каждого запроса-треугольника в отдельной строке одно целое число — ответ на запрос.

Пример

stdin	stdout
5	1
1 2 2	2
1 4 4	
1 6 6	
2 1 1 2	
2 1 1 6	



Подсказка по решению

Есть `set<int>` с операциями `order_of_key` и `find_by_order`. Называется `tree<int>`. Разобраться, как им пользоваться, можно, прочитав [пост на codeforces](#).

Для мастеров AI

Правила.

Это блок задач про промтинг. Пользоваться можно только бесплатными версиями ИИ. В шапке исходника указывать последовательность промтов, и сайты, куда они были отправлены. Если в процессе использования ИИ вы получили какие-то важные идеи для решения, это тоже часть решения. Если с помощью ИИ вы написали генератор тестов или стресс-тест, это тоже следует указывать. Задокументируйте, пожалуйста, сделанную работу в шапке отосланного решения. Если это нужно, вы можете писать часть кода сами.

Задача 28J. Фыр-Фыр-Фенвик [0.55 сек, 128 mb]

Даны n точек с весами на плоскости. Каждая задаётся тремя числами x_i, y_i, w_i (координаты и вес). Вам нужно обработать m запросов двух типов:

- `get rx ry` – посчитать сумму весов точек, у которых $x_i \leq rx$ и $y_i \leq ry$.
- `change i z` – задать i -й точке новый вес равный z .

Формат входных данных

На первой строке число n ($1 \leq n \leq 100\,000$). На следующих n строках тройки целых чисел x_i, y_i, w_i ($0 \leq x_i, y_i, w_i < 10^9$). Следующая строка содержит количество запросов m ($1 \leq m \leq 300\,000$). На следующих m строках описания запросов в формате `get rx, ry` и `change i z`. Здесь $1 \leq i \leq n$, а остальные числа целые от 0 до 10^9-1 .

Формат выходных данных

Для каждого запроса типа “get” выведите одно целое число на отдельной строке — ответ на запрос.

Пример

stdin	stdout
3	110
1 1 10	1210
2 3 100	1010
3 2 1000	
4	
get 2 3	
change 2 200	
get 3 3	
get 3 2	

Подсказка по решению

Это задача на тему «Фенвик Фенвиков». Возможно, вы не знаете Фенвика, это нормально. AI в помощь. Задача оптимизационная, важно следить за константой.