

Содержание

Задачи про Дерево Отрезков	3
Задача 18.A. Сумма [0.2 sec, 256 mb]	3
Задача 18.B. RMQ [0.4 sec, 256 mb]	4
Задача 18.C. Художник [1 sec, 256 mb]	5
Задачи про Динамическое Дерево Отрезков	6
Задача 18.D. Минимум и максимум [0.2 sec, 256 mb]	6
Задачи про ScanLine	7
Задача 18.E. Звёзды [0.15 sec, 256 mb]	7
Задача 18.F. Окна [1 sec, 256 mb]	8
Задача 18.G. Прямоугольники [1 sec, 256 mb]	9
Бонус	10
Задача 18.H. Gyakkyou Buraі Kaiji [5 sec, 256 mb]	10
Задача 18.I. Различные числа [1.5 sec, 256 mb]	12
Задача 18.J. Треугольник [2 sec, 256 mb]	13

Пример работы с файлами.

Если вы не умеете читать/выводить данные, или открывать файлы, воспользуйтесь примерами. <http://acm.math.spbu.ru/~sk1/algo/sum/>

Пример работы с файлами.

В некоторых задачах большой ввод и вывод. Про ввод-вывод в C++:

http://acm.math.spbu.ru/~sk1/algo/input-output/cpp_common.html

Имеет смысл пользоваться супер быстрым вводом-выводом. Две версии:

http://acm.math.spbu.ru/~sk1/algo/input-output/io_export.cpp.html

http://acm.math.spbu.ru/~sk1/algo/input-output/fread_write_export.cpp.html

Выделение памяти.

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

<http://acm.math.spbu.ru/~sk1/algo/memory.cpp.html>

Задачи про Дерево Отрезков

В этих задачах нужно использовать только дерево отрезков.

Можно *дерево Фенвика*

Задача 18.A. Сумма [0.2 sec, 256 mb]

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	

Задача 18.В. RMQ [0.4 sec, 256 mb]

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

- Запрос “max $l r$ ”. Требуется найти максимум в массиве a от l -ой ячейки до r -ой включительно.
- Запрос “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 “max $l r$ ”, либо “add $l r v$ ”. При этом $1 \leq l \leq r \leq n$, $|v| \leq 10^5$.

Формат выходных данных

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

Примеры

rmq.in	rmq.out
5 3	3
1 2 3 4 -5	7
max 1 3	
add 1 2 5	
max 1 3	

Задача 18.C. Художник [1 сек, 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

painter.in	painter.out
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	

Задачи про Динамическое Дерево Отрезков

В этой задаче обязательно использовать только дерево отрезков.

Задача 18.D. Минимум и максимум [0.2 сек, 256 mb]

Пусть есть мультимножество целых чисел (множество с возможными повторениями). Необходимо реализовать структуру данных для их хранения, поддерживающую следующие операции: `GetMin` — извлечение минимума, `GetMax` — извлечение максимума, `Insert(N)` — добавление числа в множество.

Формат входных данных

В первой строке входного файла записано одно целое число N ($1 \leq N \leq 100\,000$) — число запросов к структуре. Затем в N строках следуют запросы по одному в строке: `GetMin`, `GetMax`, `Insert(A)` — извлечение минимума, максимума и добавление числа A ($1 \leq A \leq 2^{31} - 1$). Запросы корректны, то есть нет операций извлечения для пустого множества.

Формат выходных данных

Для каждого запроса `GetMin` или `GetMax` выведите то число, которое было извлечено.

Примеры

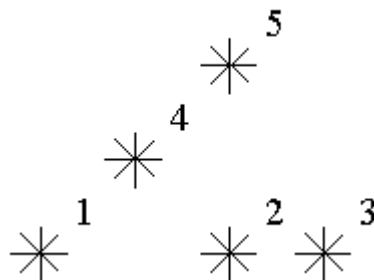
minmax.in	minmax.out
10	1
Insert(100)	100
Insert(99)	1
Insert(1)	2
Insert(2)	99
GetMin	
GetMax	
Insert(1)	
GetMin	
GetMin	
GetMax	

Задачи про ScanLine

В этих задачах обязательно написать решение за $\mathcal{O}(n \log n)$.

Задача 18.E. Звёзды [0.15 сек, 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

stars.in	stars.out
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	

Задача 18.F. Окна [1 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 $(x_{(1,i)}, y_{(1,i)})$ — координаты левого верхнего угла i -го окна, а $(x_{(2,i)}, y_{(2,i)})$ — правого нижнего (на экране компьютера y растет 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

windows.in	windows.out
2	2
0 0 3 3	3 2
1 1 4 4	

Задача 18.G. Прямоугольники [1 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

rects.in	rects.out
4	10
1 1 2 2 6	3 2
3 1 4 2 5	
0 3 1 4 5	
5 1 6 2 4	

Бонус

Задача 18.Н. Gyakkyou Burai Kaiji [5 sec, 256 mb]

You shouldn't let kings like myself draw twice.

Однажды, прежде чем появится здесь. Каиджи потерял все. Единственное, что у него осталось — жалкая жизнь.



Правила этой игры практически такие же. Есть N различных типов карт, все типы пронумерованы числами от 1 до N включительно. Каиджи хранит свои карты в колодах. Карты одинакового типа он кладет в одинаковые колоды, а карты разного типа в разные. Индекс каждой колоды совпадает с индексом типа карт, которые она содержит.

В любой момент времени, у Каиджи может быть от 0 до 999 999 999 карт каждого типа. Однако, сейчас игроки не могут купить, продать или обменяться картами. Таким образом, количество карт каждого типа, которое есть у Каиджи, остается одинаковым в течение всей игры. В течение хода, Каиджи может сыграть, используя только одну колоду с индексами из отрезка $[i, j]$ где i и j параметры хода.

Каиджи уже изучил поведение и стратегии всех игроков и разработал выигрышную стратегию. Теперь все, что ему надо, это быстро находить ответы к текущему типу вопроса: на ходе с параметрами i и j , какое количество карт в k -й по величине колоде среди колод, которые он использует? Помогите ему ответить на эти вопросы.

Первая строка содержит целое число N , количество типов карт ($1 \leq N \leq 450\,000$).

Вторая строка используется, чтобы сгенерировать целые числа a_i , начальное количество карт каждого типа, которое есть у Каиджи ($0 \leq a_i < 10^9$). Она содержит три целых числа a_1 , l и m . ($0 \leq a_1, l, m < 10^9$); $2 \leq i \leq N$,

$$a_i = (a_{i-1} \cdot l + m) \bmod 10^9.$$

Третья строка содержит целое число B — число противников ($1 \leq B \leq 1000$). В следующих строках описывают множество игр с отдельным противником. Каждое множество описывается десятью целыми числами. Первым идет число G — число игр, сыгранных с этим противником. 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 -й игре с текущим противником, Каиджи хочет знать количество карт в k_g -й по величине колоде среди всех колод с индексами из отрезка $[i_g, j_g]$. Общее количество игр, сыгранных Каиджи, не превышает 600 000.

Формат выходных данных

Для каждой игры g с каждым противником b , найдите число карт в k_g -й по величине колоде, среди его колод с индексами из отрезка $[i_g, j_g]$. Выведите одно число: сумму всех этих значений.

Пример

gyakkyou.in	gyakkyou.out
5	15
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	

У Каиджи есть i карт i -го типа для всех $i = 1, 2, 3, 4, 5$. Каждый тип выбирается только один раз. Таким образом ответ 15.

Задача 18.1. Различные числа [1.5 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

distinctnum.in	distinctnum.out
5	3
1 1 2 1 3	2
3	3
1 5	
2 4	
3 5	

Задача 18J. Треугольник [2 сек, 256 mb]

Ваша задача — написать программу, хранящую множество 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

triangle.in	triangle.out
5	1
1 2 2	2
1 4 4	
1 6 6	
2 1 1 2	
2 1 1 6	

Замечание

