

Содержание

Задачи про STL	2
Задача А. Миллиардеры [0.4 sec, 256 mb]	2
Задача В. Минимум и максимум [0.15 sec, 256 mb]	3
Задачи про Дерево Отрезков	4
Задача С. Сумма [0.2 sec, 256 mb]	4
Задача D. RMQ [0.5 sec, 256 mb]	5
Задача Е. Художник [1 sec, 256 mb]	6
Задачи про ScanLine	7
Задача F. Stars [0.2 sec, 256 mb]	7
Задача G. Окна [1 sec, 256 mb]	8
Задача H. Прямоугольники [1 sec, 256 mb]	9
Бонус	10
Задача I. Gyakkyou Burai Kaiji [5 sec, 256 mb]	10
Задача J. Различные числа [1.5 sec, 256 mb]	12

В некоторых задачах большой ввод и вывод. Имеет смысл пользоваться супер быстрым вводом-выводом:

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

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 STL

В этих задачах обязательно использовать только стандартные STL-контейнеры.

Задача А. Миллиардеры [0.4 сек, 256 mb]

Возможно, вы знаете, что из всех городов мира больше всего миллиардеров живёт в Москве. Но, поскольку работа миллиардера подразумевает частые перемещения по всему свету, в определённые дни какой-то другой город может занимать первую строчку в таком рейтинге. Ваши приятели из ФСБ, ФБР, MI5 и Шин Бет скинули вам списки перемещений всех миллиардеров за последнее время. Ваш работодатель просит посчитать, сколько дней в течение этого периода каждый из городов мира был первым по общей сумме денег миллиардеров, находящихся в нём.

Формат входных данных

В первой строке записано число n — количество миллиардеров ($1 \leq n \leq 10\,000$). Каждая из следующих n строк содержит данные на определённого человека: его имя, название города, где он находился в первый день данного периода, и размер состояния. В следующей строке записаны два числа: m — количество дней, о которых есть данные ($1 \leq m \leq 50\,000$), k — количество зарегистрированных перемещений миллиардеров ($0 \leq k \leq 50\,000$). Следующие k строк содержат список перемещений в формате: номер дня (от 1 до $m-1$), имя человека, название города назначения. Вы можете считать, что миллиардеры путешествуют не чаще одного раза в день, и что они отбывают поздно вечером и прибывают в город назначения рано утром следующего дня. Список упорядочен по возрастанию номера дня. Все имена и названия городов состоят не более чем из 20 латинских букв, регистр букв имеет значение. Состояния миллиардеров лежат в пределах от 1 до 100 миллиардов.

Формат выходных данных

В каждой строке должно содержаться название города и, через пробел, количество дней, в течение которых этот город лидировал по общему состоянию миллиардеров, находящихся в нём. Если таких дней не было, пропустите этот город. Города должны быть отсортированы по алфавиту (используйте обычный порядок символов: ABC...Zabc...z).

Примеры

millioner.in	millioner.out
5	Anadyr 5
Abramovich London 15000000000	London 14
Deripaska Moscow 10000000000	Moscow 1
Potantin Moscow 5000000000	
Berezovsky London 2500000000	
Khodorkovsky Chita 10000000000	
25 9	
1 Abramovich Anadyr	
5 Potantin Courchevel	
10 Abramovich Moscow	
11 Abramovich London	
11 Deripaska StPetersburg	
15 Potantin Norilsk	
20 Berezovsky Tbilisi	
21 Potantin StPetersburg	
22 Berezovsky London	

Задача В. Минимум и максимум [0.15 sec, 256 mb]

Пусть есть множество целых чисел. 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 Дерево Отрезков

В этих задачах обязательно использовать только дерево отрезков.

Задача С. 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 D. RMQ [0.5 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 E. Художник [1 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 интересующие 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

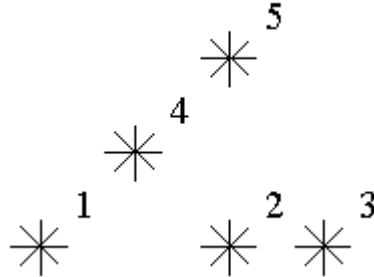
painter.in	painter.out
7	0 0
W 2 3	1 2
B 2 2	1 4
B 4 2	1 4
B 3 2	2 6
B 7 2	3 5
W 3 1	0 0
W 0 10	

Задачи про ScanLine

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

Задача F. Stars [0.2 sec, 256 mb]

Astronomers often examine star maps where stars are represented by points on a plane and each star has Cartesian coordinates. Let the level of a star be an amount of the stars that are not higher and not to the right of the given star. Astronomers want to know the distribution of the levels of the stars.



For example, look at the map shown on the figure above. Level of the star number 5 is equal to 3 (it's formed by three stars with a numbers 1, 2 and 4). And the levels of the stars numbered by 2 and 4 are 1. At this map there are only one star of the level 0, two stars of the level 1, one star of the level 2, and one star of the level 3. You are to write a program that will count the amounts of the stars of each level on a given map.

Формат входных данных

Input contains one or more tests. Each test is described as follows.

The first line of the test contains a number of stars N ($1 \leq N \leq 15\,000$). The following N lines describe coordinates of stars (two integers X and Y per line separated by a space, $0 \leq X, Y \leq 32\,000$). There can be only one star at one point of the plane. Stars are listed in ascending order of Y coordinate. Stars with equal Y coordinates are listed in ascending order of X coordinate.

Формат выходных данных

Output should contain answers to all tests. Each answer should be described as follows.

N lines, one number per line. The first line contains amount of stars of the level 0, the second does amount of stars of the level 1 and so on, the last line contains amount of stars of the level $N-1$.

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 G. 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 Н. Прямоугольники [1 сек, 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 I. 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 J. Различные числа [1.5 сек, 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	