Содержание

Вы не умеете читать/выводить данные, открывать файлы? Воспользуйтесь примерами.

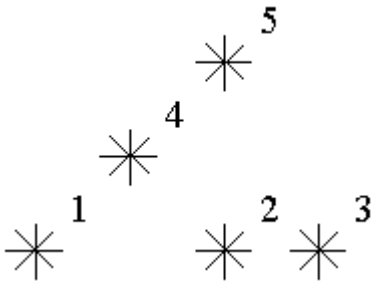
В некоторых задачах большой ввод и вывод. Пользуйтесь быстрым вводом-выводом.

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

Must have

Задача A. Звёзды [1, 512]

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 ($1 \leqslant N \leqslant 15\,000$). Следующие N строк описывают координаты звёзд (два целых числа X и Y, разделённые пробелом, $0 \leqslant X, Y \leqslant 32,000$). 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 ($i = 0 \ldots N-1$).

Примеры

stars.in	stars.out
5 1 1 5 1 7 1 3 3 5 5 5 1 1 5 1 7 1 3 3 5 5	1 2 1 1 0 1 2 1 1 0

Замечание

Простейший scanline.

Обязательные задачи

Задача B. Окна [1, 512]

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 ($1 \leqslant n \leqslant 50\,000$).

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 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 0 0 3 3 1 1 4 4	2 3 2

Задача C. Перестановки [1, 512]

Вася выписал на доске в каком-то порядке все числа от 1 по N, каждое число ровно по одному разу. Количество чисел оказалось довольно большим, поэтому Вася не может окинуть взглядом все числа. Однако ему надо всё-таки представлять эту последовательность, поэтому он написал программу, которая отвечает на вопрос — сколько среди чисел, стоящих на позициях с x по y, по величине лежат в интервале от k до l. Сделайте то же самое.

Формат входных данных

В первой строке лежит два натуральных числа — $1 \leqslant N \leqslant 100\,000$ — количество чисел, которые выписал Вася и $1 \leqslant M \leqslant 100\,000$ — количество вопросов, которые Вася хочет задать программе. Во второй строке дано N чисел — последовательность чисел, выписанных Васей. Далее в M строках находятся описания вопросов. Каждая строка содержит четыре целых числа $1 \leqslant x \leqslant y \leqslant N$ и $1 \leqslant k \leqslant l \leqslant N$.

Формат выходных данных

Выведите M строк, каждая должна содержать единственное число — ответ на Васин вопрос.

Пример

permutation.in	permutation.out
4 2 1 2 3 4 1 2 2 3 1 3 1 3	1 3

Задача D. Persistent Array [1, 512]

Дан массив (вернее, первая, начальная его версия).
Нужно уметь в online отвечать на два запроса:

• `create i j x` — создать из i-й версии новую, в которой j-й элемент равен x, а остальные элементы такие же, как в i-й версии.
• `get i j` — сказать, чему равен j-й элемент в i-й версии.

Это интерактивная задача. Запросы нужно обрабатывать в online, результат каждого запроса `flush`-ить до чтения следующего.

Формат входных данных

Количество чисел в массиве N ($1 \leqslant N \leqslant 10^5$) и N элементов массива. Далее количество запросов M ($1 \leqslant M \leqslant 10^5$) и M запросов. Формат описания запросов можно посмотреть в примере. Если уже существует K версий, новая версия получает номер $K + 1$. И исходные, и новые элементы массива — целые числа от 0 до 10^9. Элементы в массиве нумеруются числами от 1 до N.

Формат выходных данных

На каждый запрос типа `get` вывести соответствующий элемент нужного массива.

Пример

parray.in	parray.out
6 1 2 3 4 5 6 11 create 1 6 10 create 2 5 8 create 1 5 30 get 1 6 get 1 5 get 2 6 get 2 5 get 3 6 get 3 5 get 4 6 get 4 5	6 5 10 5 10 8 6 30

Замечание

Пример корректного кода ввода/вывода:

```
string query;
for (int i = 0; i < queryCnt; i++) {
    getline(cin, query);
    int answer = process(query);
    cout << answer << endl; // endl делает flush
}
```

Задача E. CHM [1, 512]

Ваша задача — реализовать **Persistent Disjoint-Set-Union**. Что это значит?
Про **Disjoint-Set-Union**:
Изначально у вас есть n элементов. Нужно научиться отвечать на 2 типа запросов.

- `+ a b` — объединить множества, в которых лежат вершины a и b
- `? a b` — сказать, лежат ли вершины a и b сейчас в одном множестве

Про **Persistent**:
Теперь у нас будет несколько копий (версий) структуры данных **Disjoint-Set-Union**.
Запросы будут выглядеть так:

- `+ i a b` — запрос к i-й структуре, объединить множества, в которых лежат вершины a и b. При этом i-я структура остается не изменной, создается новая версия, ей присваивается новый номер (какой? читайте дальше)
- `? i a b` — запрос к i-й структуре, сказать, лежат ли вершины a и b сейчас в одном множестве

Формат входных данных

На первой строке 2 числа N ($1 \leqslant N \leqslant 10^5$) и K ($0 \leqslant K \leqslant 10^5$) — число элементов и число запросов. Изначально все элементы находятся в различных множествах. Эта изначальная копия (версия) структуры имеет номер 0.

Далее следуют K строк, на каждой описание очередного запроса. Формат запросов описан выше. Запросы нумеруются целыми числами от 1 до K.

Пусть j-й из K запросов имеет вид «`+ i a b`». Тогда новая версия получит номер j. Запросы вида «`? i a b`» не порождают новых структур.

Формат выходных данных

Для каждого запроса вида `? i a b` на отдельной строке нужно вывести `YES` или `NO`.

Пример

snm.in	snm.out
4 7 + 0 1 2 ? 0 1 2 ? 1 1 2 + 1 2 3 ? 4 3 1 ? 0 4 4 ? 4 1 4	NO YES YES YES NO

Дополнительные задачи

Задача F. Дерево [1, 512]

Задано подвешенное дерево, содержащее n ($1 \leqslant n \leqslant 1\,000\,000$) вершин. Каждая вершина покрашена в один из n цветов. Требуется для каждой вершины v вычислить количество различных цветов, встречающихся в поддереве с корнем v.

Формат входных данных

В первой строке входного файла задано число n. Последующие n строк описывают вершины, по одной в строке. Описание очередной вершины i имеет вид p_i c_i, где p_i — номер родителя вершины i, а c_i — цвет вершины i ($1 \leqslant c_i \leqslant n$). Для корня дерева $p_i = 0$.

Формат выходных данных

Выведите n чисел, обозначающих количества различных цветов в поддеревьях с корнями в вершинах $1, \ldots, n$.

Примеры

tree.in	tree.out
5 2 1 3 2 0 3 3 3 2 1	1 2 3 1 1

Задача G. Треугольник [1, 512]

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 ($1 \leqslant m \leqslant 100\,000$). 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 $|x|, |y|, r \leqslant 10^8$ и $r > 0$.

Формат выходных данных

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

Пример

5 1 2 2 1 4 4 1 6 6 2 1 1 2 2 1 1 6	1 2

Замечание