

Задача А. Место встречи изменить нельзя

Имя входного файла: `rendezvous.in`
Имя выходного файла: `rendezvous.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

256 мегабайт

Даны N точек. Найдите такие две из них, что расстояние между ними минимально.

Формат входных данных

Первая строка входного файла содержит целое число N ($2 \leq N \leq 100\,000$) — количество точек. Каждая из следующих N строк содержит пару целых чисел X и Y , разделённых пробелом, — координаты ($-1\,000\,000\,000 \leq X, Y \leq 1\,000\,000\,000$). Все точки различны.

Формат выходных данных

Единственная строка выходного файла должна содержать координаты двух выбранных точек.

Пример

<code>rendezvous.in</code>	<code>rendezvous.out</code>
4	0 0
0 0	0 1
0 1	
1 1	
1 0	

Задача В. Соединение и разъединение

Имя входного файла: `connect.in`
Имя выходного файла: `connect.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

512 мегабайт

Вы когда-нибудь слышали про обход в глубину? Например, используя этот алгоритм, вы можете проверить является ли граф связным за время $O(E)$. Вы можете даже посчитать количество компонент связности за то же время.

А вы когда-нибудь слышали про систему непересекающихся множеств? Используя эту структуру, вы можете быстро обрабатывать запросы “Добавить ребро в граф” и “Посчитать количество компонент связности в графе”.

А вы когда-нибудь слышали о *динамической* задаче связности? В этой задаче вам необходимо обрабатывать три типа запросов:

1. Добавить ребро в граф.
2. Удалить ребро из графа.
3. Посчитать количество компонент связности в графе.

Можно считать, что граф является неориентированным. Изначально граф является пустым.

Формат входных данных

В первой строке находятся два целых числа N и K — количество вершин и количество запросов, соответственно ($1 \leq N \leq 300\,000$, $0 \leq K \leq 300\,000$). Следующие K строк содержат запросы, по одному в строке. Каждый запрос имеет один из трех типов:

1. $+ u v$: Добавить ребро между вершинами u и v . Гарантируется, что такого ребра нет.
2. $- u v$: Удалить ребро между u и v . Гарантируется, что такое ребро есть.
3. $?$: Посчитать количество компонент связности в графе.

Вершины пронумерованы целыми числами от 1 до N . Во всех запросах $u \neq v$.

Формат выходных данных

Для каждого запроса типа ‘?’, Выведите количество компонент связности в момент запроса.

Пример

<code>connect.in</code>	<code>connect.out</code>
5 11	5
?	1
+ 1 2	1
+ 2 3	2
+ 3 4	
+ 4 5	
+ 5 1	
?	
- 2 3	
?	
- 4 5	
?	

Задача С. Большие Диаграммы Юнга

Имя входного файла: `multiply.in`
Имя выходного файла: `multiply.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

256 мегабайт

Яша и Саша — два роскошных брата, которые любят дарить друг другу подарки на день рождения. И вот, на очередной день рождения, Яша подарил Саше на день рождения Диаграмму Юнга. Саша решил посчитать число клеток в этой диаграмме. Но Саша опаздывал на матч по футболу, поэтому посчитал только число клеток в первой строке и первом столбце. И теперь ему интересно, какое максимальное по модулю число клеток может быть в этой диаграмме.

Формат входных данных

В первой строке входных данных находится одно целое число — размер первой строки диаграммы. Во второй строке находится другое целое число — размер первого столбца диаграммы. Длины этих чисел не превосходит двухсот пятидесяти тысяч символов. Размеры строк этой диаграммы могут быть отрицательными.

Формат выходных данных

В единственной строке выведите единственное число — максимальный по модулю размер диаграммы.

Пример

<code>multiply.in</code>	<code>multiply.out</code>
2	4
2	

Задача D. Прибавления на отрезках

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Гриша пришел на констест и увидел там следующую задачу.

Дан массив длины n , изначально состоящий из нулей. Элементы массива пронумерованы от 1 до n . К массиву было применено q операций. i -я операция задается тремя целыми числами l_i , r_i и x_i ($1 \leq l_i \leq r_i \leq n$), ($1 \leq x_i \leq n$) и означает, что к элементам с номерами $l_i, l_i + 1, \dots, r_i$ прибавили число x_i . Требуется найти максимум в массиве после применения всех этих операций.

Но Гриша не из глупых! Он решил эту задачу очень быстро.

Однако что-то в нем переклинило, и он задумался: «интересно, а какие значения может принять максимум в массиве после применения некоторого подмножества данных операций?».

Помогите Грише, найдите все такие целые числа y от 1 до n , что после применения некоторого (возможно, пустого) подмножества данных операций максимум в массиве равен y .

Формат входных данных

В первой строке находятся два целых числа n и q ($1 \leq n, q \leq 10^4$) — длина массива и количество запросов в исходной задаче.

В следующих q строках описаны запросы, по одному в строке. i -я из этих строк содержит три целых числа l_i , r_i и x_i ($1 \leq l_i \leq r_i \leq n$, $1 \leq x_i \leq n$), что обозначает запрос на добавление числа x_i на отрезке с l_i -го по r_i -й элемент включительно.

Формат выходных данных

В первую строку выведите единственное число k , обозначающее количество возможных целых чисел от 1 до n , которым может быть равен максимум в массиве после применения некоторого (возможно, пустого) подмножества данных операций.

В следующей строке выведите через пробел все k чисел от 1 до n — возможные значения максимума. Выводите эти числа в **возрастающем порядке**.

Примеры

стандартный ввод	стандартный вывод
4 3 1 3 1 2 4 2 3 4 4	4 1 2 3 4
7 2 1 5 1 3 7 2	3 1 2 3
10 3 1 1 2 1 1 3 1 1 6	6 2 3 5 6 8 9

Замечание

Если в первом тестовом примере оставить только первый запрос, то максимум будет равен 1. Если оставить только второй запрос, то максимум будет равен 2. Если оставить первые два запроса, то максимум будет равен 3. Если оставить только третий запрос, то максимум будет равен 4. Но если оставить третий запрос и еще какой-то, максимум будет больше n , поэтому его выводить не требуется.

Во втором тестовом примере, оставив только первый запрос, можно получить 1. Оставив только второй, можно получить 2. А если оставить все запросы, максимум будет равен 3.

В третьем тестовом примере можно получить максимумы так:

- Можно получить максимум 2 оставив запросы: (1).
- Можно получить максимум 3 оставив запросы: (2).
- Можно получить максимум 5 оставив запросы: (1, 2).
- Можно получить максимум 6 оставив запросы: (3).
- Можно получить максимум 8 оставив запросы: (1, 3).
- Можно получить максимум 9 оставив запросы: (2, 3).

Задача Е. Дружелюбные хомячки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

На плоскости живут n хомячков. Каждый в точке с целыми координатами. Хомячки дружат, если существует прямоугольник со сторонами, параллельными осям координат, содержащий этих двух хомячков и не содержащий никаких других.

Прямоугольник содержит хомячка, если точка, в которой он живет, лежит внутри прямоугольника или на его границе.

Сколько пар хомячков дружат?

Формат входных данных

На первой строке число n ($1 \leq n \leq 100\,000$).

Следующие n строк содержат по два целых числа – координаты точек, в которых живут хомячки.

Все точки различны, а координаты целые, по модулю не превосходят 10^9 .

Формат выходных данных

Выведите одно целое число – количество пар дружащих хомячков.

Пример

стандартный ввод	стандартный вывод
5 0 0 0 2 2 0 2 2 1 1	8

Задача F. Количество инверсий

Имя входного файла: `inverse.in`
Имя выходного файла: `inverse.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Напишите программу, которая для заданного массива $A = \langle a_1, a_2, \dots, a_n \rangle$ находит количество пар (i, j) таких, что $i < j$ и $a_i > a_j$. Обратите внимание на то, что ответ может не влезать в `int`.

Формат входных данных

Первая строка входного файла содержит натуральное число n ($1 \leq n \leq 100\,000$) — количество элементов массива. Вторая строка содержит n попарно различных элементов массива A — целых неотрицательных чисел, не превосходящих 10^9 .

Формат выходных данных

В выходной файл выведите одно число — ответ на задачу.

Примеры

<code>inverse.in</code>	<code>inverse.out</code>
5 6 11 18 28 31	0
8 999994 999989 999982 999972 999969 999961 999954 999950	28