

Содержание

Задачи	2
1 Задача А. Минимальное покрытие [0.5 секунд, 256 mb]	2
2 Задача В. Точки и отрезки [0.5 секунд, 256 mb]	3
3 Задача С. Обмен [0.5 секунд, 256 mb]	4
4 Задача D. Q-я порядковая статистика [0.8 секунд, 256 mb]	5
Бонус	6
5 Задача Е. Сортировка за линейное время [0.6 секунд, 256 mb]	6
6 Задача F. Минимумы в подматрицах [0.7 секунд, 256 mb]	7

В некоторых задачах большой ввод и вывод. Имеет смысл пользоваться супер быстрым вводом-выводом, например: <http://acm.math.spbu.ru/~sk1/algo/input-output/cpp.html>

Задачи

1 Задача А. Минимальное покрытие [0.5 секунд, 256 mb]

На прямой задано некоторое множество отрезков с целочисленными координатами концов $[L_i, R_i]$. Выберите среди данного множества подмножество отрезков, целиком покрывающее отрезок $[0, M]$, (M — натуральное число), содержащее наименьшее число отрезков.

Формат входных данных

В первой строке указана константа M ($1 \leq M \leq 5000$). В каждой последующей строке записана пара чисел L_i и R_i ($|L_i|, |R_i| \leq 50000$), задающая координаты левого и правого концов отрезков. Список завершается парой нулей. Общее число отрезков не превышает 100 000.

Формат выходных данных

В первой строке выходного файла выведите минимальное число отрезков, необходимое для покрытия отрезка $[0, M]$. Далее выведите список покрывающего подмножества, упорядоченный по возрастанию координат левых концов отрезков. Список отрезков выводится в том же формате, что и во входе. Завершающие два нуля выводить не нужно.

Если покрытие отрезка $[0, M]$ исходным множеством отрезков $[L_i, R_i]$ невозможно, то следует вывести единственную фразу “No solution”.

Примеры

cover.in	cover.out
1 -1 0 -5 -3 2 5 0 0	No solution
1 -1 0 0 1 0 0	1 0 1

2 Задача В. Точки и отрезки [0.5 секунд, 256 mb]

Дано n отрезков на числовой прямой и m точек на этой же прямой. Для каждой из данных точек определите, скольким отрезкам она принадлежит. Точка x считается принадлежащей отрезку с концами a и b , если выполняется двойное неравенство $\min(a, b) \leq x \leq \max(a, b)$.

Формат входных данных

Первая строка содержит два целых числа n ($1 \leq n \leq 10^5$) — число отрезков и m ($1 \leq m \leq 10^5$) — число точек. В следующих n строках записаны по два целых числа a_i и b_i — координаты концов соответствующего отрезка. В последней строке записаны m целых чисел — координаты точек. Все числа во входном файле не превосходят по модулю 10^9 .

Формат выходных данных

В выходной файл выведите m чисел — для каждой точки выведите количество отрезков, в которых она содержится.

Примеры

segments.in	segments.out
2 2 0 5 7 10 1 6	1 0
1 3 -10 10 -100 100 0	0 0 1

3 Задача С. Обмен [0.5 секунд, 256 mb]

Пусть все натуральные числа исходно организованы в список в естественном порядке. Решается выполнить следующую операцию: $swap(a, b)$. Эта операция возвращает в качестве результата расстояние в текущем списке между числами a и b и меняет их местами.

Задана последовательность операций $swap$. Требуется вывести в выходной файл результат всех этих операций.

Формат входных данных

Первая строка входного файла содержит число n ($1 \leq n \leq 200\,000$) — количество операций. Каждая из следующих n строк содержит по два числа в диапазоне от 1 до 10^9 — аргументы операций $swap$.

Формат выходных данных

Для каждой операции во входном файле выведите ее результат.

Пример

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

4 Задача D. Q-я порядковая статистика [0.8 секунд, 256 mb]

Вам дан массив из n случайных целых чисел. Ваша задача — отсортировать массив и вывести q -е число в получившемся порядке.

Формат входных данных

На первой строке числа n, q . ($1 \leq q \leq n \leq 10^7$). На второй строке пара целых чисел a, b от 1 до 10^9 , используемая в генераторе случайных чисел.

```
1. unsigned int cur = 0; // беззнаковое 32-битное число
2. unsigned int nextRand24() {
3.     cur = cur * a + b; // вычисляется с переполнениями
4.     return cur >> 8; // число от 0 до  $2^{24} - 1$ .
5. }
6. unsigned int nextRand32() {
7.     unsigned int a = nextRand24(), b = nextRand24();
8.     return (a << 8) ^ b; // число от 0 до  $2^{32} - 1$ .
9. }
```

Элементы массива генерируются последовательно. $x_i = \text{nextRand32}()$;

Формат выходных данных

Выведите ответ на запрос.

Примеры

qstat.in	qstat.out
6 3 239 13	197852696

Замечание

Сгенерированный массив: 12, 130926, 3941054950, 2013898548, 197852696, 2753287507.
Внимание: нельзя пользоваться стандартными функциями, например, `nth_element`.

Бонус

5 Задача E. Сортировка за линейное время [0.6 секунд, 256 mb]

Дан массив случайных целых чисел, нужно отсортировать его.

Формат входных данных

На первой строке количество тестов t ($1 \leq t \leq 200$) и число n ($1 \leq n \leq 50\,000$) — размер массива в каждом из тестов. На второй строке пара целых чисел a, b от 1 до 10^9 , используемая в генераторе случайных чисел.

```
1. unsigned int cur = 0; // беззнаковое 32-битное число
2. unsigned int nextRand24() {
3.     cur = cur * a + b; // вычисляется с переполнениями
4.     return cur >> 8; // число от 0 до  $2^{24} - 1$ .
5. }
6. unsigned int nextRand32() {
7.     unsigned int a = nextRand24(), b = nextRand24();
8.     return (a << 8) ^ b; // число от 0 до  $2^{32} - 1$ .
9. }
```

Тесты генерируются последовательно.

Элементы массива генерируются последовательно. $x_i = \text{nextRand32}()$;

Формат выходных данных

Для каждого теста выведите на отдельной строке $\sum_{i=1}^n x_i \cdot i$.

Примеры

buckets.in	buckets.out
1 6	8906224639
239 13	

Замечание

Сгенерированный массив: 12, 130926, 3941054950, 2013898548, 197852696, 2753287507.

В этой задаче очень небольшой запас по времени. Если она не сдается, это нормально.

6 Задача F. Минимумы в подматрицах [0.7 секунд, 256 mb]

Дана матрица $n \times n$, состоящая из целых чисел. Для каждой её подматрицы размера $L \times L$ найдите минимум в этой подматрице. Подматрицей здесь называется “подпрямоугольник”.

Внимание. Решение должно работать за $O(n^2)$.

Формат входных данных

Первая строка входных данных содержит два целых числа n и L ($1 \leq L \leq n \leq 1000$). Далее в n строках идет описание матрицы $n \times n$, по n чисел в каждой строке. Все числа в матрицы целые, от -10^9 до 10^9 .

Формат выходных данных

Выведите $n - L + 1$ строку по $n - L + 1$ числу в каждой. j -е число в i -й строке должно быть равно минимуму в подматрице размера $L \times L$ с левым верхним углом на пересечении i -й строки и j -го столбца исходной матрицы.

Примеры

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