**Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики**

**Алгоритмы и структуры данных**

**Лабораторная работа №3(4)**

**Введение**

Выполнил:

Бараканов Жаргал Мырзабекович

Факультет ИКТ

Группа K3121

Преподаватель:

Харьковская Татьяна Александровна

**Санкт-Петербург**

**09.11.2021**

**Задание 1.**

1. Используя псевдокод процедуры Randomized - QuickSort, а так же Partition из презентации к Лекции 3 (страницы 8 и 12), напишите программу быстрой сортировки на Python и проверьте ее, создав несколько рандомных массивов, подходящих под параметры:

* Формат входного файла (input.txt). В первой строке входного файла содержится число n (1 ≤ n ≤ 104) — число элементов в массиве. Во второй строке находятся n различных целых чисел, по модулю не превосходящих 109.
* Формат выходного файла (output.txt). Одна строка выходного файла с отсортированным массивом. Между любыми двумя числами должен стоять ровно один пробел.
* Ограничение по времени. 2 сек.
* Ограничение по памяти. 256 мб.

2. Основное задание. Цель задачи - переделать данную реализацию рандомизированного алгоритма быстрой сортировки, чтобы она работала быстро даже с последовательностями, содержащими много одинаковых элементов. Чтобы заставить алгоритм быстрой сортировки эффективно обрабатывать последовательности с несколькими уникальными элементами, нужно заменить двухстороннее разделение на трехстороннее (смотри в Лекции 3 слайд 17). То есть ваша новая процедура разделения должна разбить массив на три части:

* A[k] < x для всех l + 1 ≤ k ≤ m1 − 1
* A[k] = x для всех m1 ≤ k ≤ m2
* A[k] > x для всех m2 + 1 ≤ k ≤ r
* Формат входного и выходного файла аналогичен п.1.

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

**Задание 3.**

«Сортировка пугалом» — это давно забытая народная потешка. Участнику под верхнюю одежду продевают деревянную палку, так что у него оказываются растопырены руки, как у огородного пугала. Перед ним ставятся n матрёшек в ряд. Из-за палки единственное, что он может сделать — это взять в руки две матрешки на расстоянии k друг от друга (то есть i-ую и i + k-ую), развернуться и поставить их обратно в ряд, таким образом поменяв их местами. Задача участника — расположить матрёшки по неубыванию размера. Может ли он это сделать?

* Формат входного файла (input.txt). В первой строчке содержатся числа n и k (1 ≤ n, k ≤ 105) – число матрёшек и размах рук. Во второй строчке содержится n целых чисел, которые по модулю не превосходят 109 – размеры матрёшек.
* Формат выходного файла (output.txt). Выведите «ДА», если возможно отсортировать матрёшки по неубыванию размера, и «НЕТ» в противном случае.
* Ограничение по времени. 2 сек.
* Ограничение по памяти. 256 мб.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст, внешний, табличка

Автоматически созданное описание

**Задание 4.**

Вам дается набор точек и набор отрезков. Цель состоит в том, чтобы вычислить для каждой точки количество отрезков, содержащих эту точку.

* Формат входного файла (input.txt). Первая строка содержит два неотрицательных целых числа s и p. s - количество отрезков, p – количество точек. Следующие s строк содержат 2 целых числа ai, bi, которые определяют i-ый отрезок [ai, bi]. Последняя строка определяет p целых чисел – точек x1, x2, ..., xp. Ограничения: 1 ≤ s, p ≤ 50000; −108 ≤ ai ≤ bi ≤ 108 для всех 0 ≤ i < s; −108 ≤ xi ≤ 108 для всех 0 ≤ j < p.
* Формат выходного файла (output.txt). Выведите p неотрицательных целых чисел k0, k1..., kp−1, где ki - это число отрезков, которые содержат xi. То есть,

ki = |j : aj ≤ xi ≤ bj |.

Изображение выглядит как текст, внешний, табличка

Автоматически созданное описание

Изображение выглядит как текст, внешний, черный, табличка

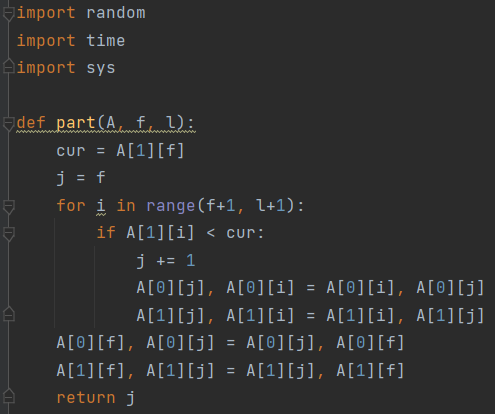
Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

**Задание 7.**

* Дано n строк, выведите их порядок после k фаз цифровой сортировки.
* Формат входного файла (input.txt). В первой строке входного файла содержатся числа n - число строк, m - их длина и k - число фаз цифровой сортировки (1 ≤ n ≤ 106, 1 ≤ k ≤ m ≤ 106, n · m ≤ 5 · 107). Далее находится описание строк, но в нетривиальном формате. Так, i-ая строка (1 ≤ i ≤ n) записана в i-ых символах второй, ..., (m + 1)-ой строк входного файла. Иными словами, строки написаны по вертикали. Это сделано специально, чтобы сортировка занимала меньше времени.
* Строки состоят из строчных латинских букв: от символа "a" до символа "z" включительно. В таблице символов ASCII все эти буквы располагаются подряд и в алфавитном порядке, код буквы "a" равен 97, код буквы "z" равен 122.
* Формат выходного файла (output.txt). Выведите номера строк в том порядке, в котором они будут после k фаз цифровой сортировки.
* Ограничение по времени. 3 сек.
* Ограничение по памяти. 256 мб.



Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст, внешний, табличка

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

**Задание 8.**

В этой задаче, ваша цель - найти K ближайших точек к началу координат среди данных n точек.

Заданы n точек на поверхности, найти K точек, которые находятся ближе к началу координат (0, 0), т.е. имеют наименьшее расстояние до начала координат. Напомним, что расстояние между двумя точками (x1, y1) и (x2, y2) равно p (x1 − x2) 2 + (y1 − y2) 2.

* Формат ввода или входного файла (input.txt). Первая строка содержит n - общее количество точек на плоскости и через пробел K - количество ближайших точек к началу координат, которые надо найти. Каждая следующая из n строк содержит 2 целых числа xi , yi , определяющие точку (xi, yi). Ограничения: 1 ≤ n ≤ 105; −109 ≤ xi , yi ≤ 109 - целые числа.
* Формат выхода или выходного файла (output.txt). Выведите K ближайших точек к началу координат в строчку в квадратных скобках через запятую. Ответ вывести в порядке возрастания расстояния до начала координат. Если оно равно, порядок произвольный.
* Ограничение по времени. 10 сек.
* Ограничение по памяти. 256 мб.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание