Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники

**Лабораторная работа №2**по курсу «Алгоритмы и структуры данных»

Выполнил:

Студент группы P3212  
Анищенко Анатолий Алексеевич

Преподаватель:

Косяков М.С.

г. Санкт-Петербург  
2020 г.

**Задача №1207 «Медиана на плоскости»**

**Пояснение к примененному алгоритму:**

Для того, чтобы разделить на 2 части плоскость найдем самую правую точку. Узнаем угол до каждой из оставшихся точек и отсортируем эти значения по величине угла. В таком случае мы сможем найти ту точку, через которую можно провести прямую, чтобы разделить плоскость так, чтобы половина оставшихся точек находились выше этой прямой, а вторая половина – ниже. Это возможно сделать из-за того, что никакие три точки не лежат на одной прямой. Время работы n \* log(n)

Второй способ решения – это взять рандомную точку и перебрать все прямые через неё и одну из оставшихся и считать кол-во точек с разных сторон. Так как ограничение всего 10000, то n^2 заходит.

**Задача №1322 «Шпион»**

**Пояснение к примененному алгоритму:**

Данная задача является реализацией обратного преобразования Берруоза-Виллера.

Для решения будем постепенно восстанавливать нашу исходную строку, прибавляя каждый раз по столбцу. Мы знаем последний столбец циклических сдвигов отсортированных, а также мы знаем первый столбец (отсортированный последний столбец). Циклически сдвинем полученные строки вправо и отсортируем полученные строки, не поверите, теперь мы знаем первые два символа отсортированных циклических сдвигов. Повторяя данный алгоритм k раз получим строку целиком. Но это долго n^2 \* log(n).

Заметим, что сортировка всегда будет одинаковой, а значит можно просто запомнить куда какой элемент надо поставить и просто переставлять элементы. Данное решение будет работать за n^2 + log(n) \* n = n^2, тоже долго.

Заметим, что нам нужна всего одна строка, а мы ещё и все её циклические сдвиги получаем. Давайте будем строить только одну строку. Это будет работать за n + n \* log(n) = n \* log(n). Или если нам известен алфавит, то мы можем использовать сортировку подсчётом, которая позволит нам оптимизировать алгоритм до времени O(n + m), где m размер алфавита.

**Задача №1604 «В стране дураков»**

**Пояснение к примененному алгоритму:**

Отсортируем массив знаки по количеству. Если максимальное кол-во знаков больше половины суммарного кол-во знаков сначала расставим через одно свободное место остальные знаки начиная с первой позиции (чтобы было как можно больше смен), а потом оставшиеся места забьём «первым» знаком. Иначе будем по порядку расставлять знаки через одно свободное место. Во втором случае каждый раз будет происходить смена, т. к. каждого знака меньше половины.

**Задача №1444 «Накормить элефпотама»**

**Пояснение к примененному алгоритму:**

Для начала нужно понять, что всегда можно обойти все тыквы. Как это достигается? Найдем угол до каждой тыквы от начальной. Отсортируем углы. При одинаковом угле сначала элефпотам должен посетить ту, которая ближе к нему, потом по прямой дойти до остальных. Т. к. все тыквы отсортированы по величине угла от начальной, то пока элефпотам идет от текущей до следующей тыквы, он не пересечет свои следы (т. к. элефпотам еще ни разу не был ни в одной из точек плоскости, угол которой расположен между этими тыквами)

Важно проследить чтобы разность между соседними углами лучей из начальной точки всегда была меньше 180ти градусов. Такой угол может быть максимум один и если он есть, то надо начинать так чтобы он не входил в наш путь.

**Задача №1726 «Кто ходит в гости…»**

**Пояснение к примененному алгоритму:**

Заметим, что можно отдельно посчитать расстояние по x и по y, т. к. путь/расстояние манхетоновское. Для нахождения найдём сумму расстояний между всеми домами и поделим её на количество” путей”.

Сохраним все координаты и отсортируем. Между каждой парой соседних домов узнаем расстояние r (по одной из координат) и узнаем какое количество человек по ней проходит (а именно i человек справа и (n - i) слева). Следовательно, суммарно по этой тропинке пройдут r \* i \* (n - i)

В данной задачи надо внимательно следить за, переполнением int’а, т. к. из-за приоритета операций может получиться неверный ответ.

**Исходный код:**

<https://github.com/AnatoliiAnishchenko/Algo_and_DS/tree/master/Part2>