

Федеральное государственное автономное образовательное учреждение высшего
образования
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»
Кафедра Вычислительной Техники

Алгоритмы и структуры данных
Задание №2

Выполнила: Калугина Марина
Группа: Р3202

г. Санкт-Петербург
2019 г.

1207. Медиана на плоскости.

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 точек находилась выше этой прямой и точки, а вторая половина - ниже.

Исходный код:

https://github.com/KaluginaMarina/algorithms_and_data_structures/blob/master/1207.cpp

1322. Шпион.

Данная задача решается при помощи преобразования Берруоза-Виллера.

Данный товарищ вводит "магический вектор T", который строится, основываясь на весах встречаемых букв, а именно:

$$T[i] = D(Li) + P(i) + 1,$$

где D(a) - сколько раз встречаются все буквы, идущие до a в алфавитном порядке, P(i) - сколько раз буква на позиции i встречается выше по столбцу.

Почему это работает? Нам нужно узнать, на каком месте в отсортированном массиве находится сдвиг, предыдущий к Mi. Он начинается с Li, т.е. стоит после D(Li) сдвигов, которые начинаются на меньшие буквы. Среди сдвигов она находится на P(i) + 1 месте.

Как узнать текст с помощью этого вектора? Мы знаем последний столбец => легко можем найти последнюю букву текста. При помощи вектора перестановки читаем следующую букву LT[i]

Исходный код:

https://github.com/KaluginaMarina/algorithms_and_data_structures/blob/master/1322.cpp

1444. Накормить эльфпотама.

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 эльфпотам 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 эльфпотам идет от текущей до следующей тыквы, он не пересечет свои следы (т.к. эльфпотам еще ни разу не был ни в одной из точек плоскости, угол которой расположен между этими тыквами)

Исходный код:

https://github.com/KaluginaMarina/algorithms_and_data_structures/blob/master/1444.cpp

1604. В стране дураков.

Отсортируем массив.

Каждый раз будем работать с двумя максимальными по количеству элементами. На каждой итерации выводим номера первых двух знаков в массиве, уменьшаем сумму знаков этих типов. В случае, если количество знаков в первых 2 элементах массива больше не максимальное восстанавливаем упорядоченность массива, перемещая эти элементы в правильное место.

Таким образом мы каждый раз ставим разные знаки до тех пор, пока не останутся знаки одного типа, для которых нельзя поставить пару.

Исходный код:

https://github.com/KaluginaMarina/algorithms_and_data_structures/blob/master/1604.cpp

1726. Кто ходит в гости...

Т.к. люди ходят только параллельно оси ox и oy , то результат - это сумма расстояний между каждым домом с каждым по x и y координатам.

Сохраним все координаты и отсортируем.

Между каждой парой соседних домов узнаем расстояние g и узнаем какое количество человек по ней проходит (а именно i человек справа и $(n - i)$ слева).

Следовательно, суммарно по этой тропинке пройдут $g * i * (n - i)$

Итоговую сумму разделим на количество тропинок между домами для получения среднего результата.

Исходный код:

https://github.com/KaluginaMarina/algorithms_and_data_structures/blob/master/1726.cpp