

Лабораторная работа № 8 по курсу дискретного анализа: Жадные алгоритмы

Выполнила студентка группы 08-308 МАИ *Шевлякова София*.

Условие

Дана последовательность длины N из целых чисел 1, 2, 3. Необходимо найти минимальное количество обменов элементов последовательности, в результате которых последовательность стала бы отсортированной.

Формат ввода: число N на первой строке и N чисел на второй строке.

Формат вывода: минимальное количество обменов.

Метод решения

Жадный алгоритм - это алгоритм, который на каждом шаге делает локально наилучший выбор в надежде на то, что итоговое решение будет оптимальным. Данная задача как раз-таки решается с помощью жадного алгоритма.

Идея решения заключается в том, что сначала мы считываем входную последовательность и подсчитываем количество единиц, двоек и троек. Если мы знаем количество каждого символа, то по этим значениям мы точно можем определить, как будет выглядеть отсортированная последовательность, можем обозначить границы блоков единиц, двоек и троек. Далее второй раз проходимся по массиву и считаем, сколько единиц находится в блоке единиц, сколько единиц находится в блоке двоек и сколько единиц находится в блоке троек, и так для каждого числа из нашего алфавита.

Чтобы избежать лишних обменов, элементы, которые уже находятся в своем блоке трогать не будем. Этот алгоритм является жадным, потому что мы предполагаем, что делая обмен с элементом, который находится правее нас, мы получим оптимальное решение.

Описание программы

В данной программе содержится один файл `main.cpp`, в котором реализован жадный алгоритм поиска оптимального решения.

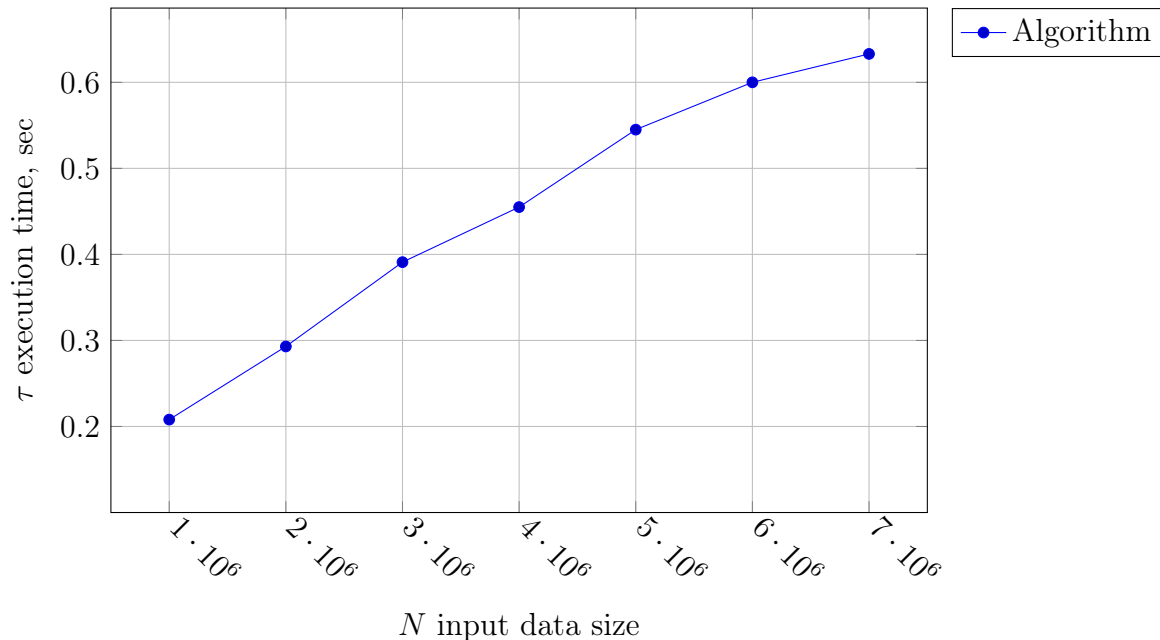
Основные этапы работы программы:

- Определим три переменные: `count_one`, `count_two`, `count_three` - для того, чтобы подсчитать количество единиц, двоек и троек. Этот шаг необходим для определения границ каждого блока.
- Определим по три переменные для единиц: `count_one_in_one`, `count_one_in_two`, `count_one_in_three`, которые советуют количеству единиц, находящихся в блоке единиц, в блоке двое и в блоке троек соответственно. Аналогично определяем

значения этих переменных для двоек и троек, все это делается за один проход по массиву.

- Сначала переместим двойки и тройки, которые находятся в блоке единиц.
- Далее делаем обмен для всех троек, которые находятся в блоке двоек, единицы там оказаться не могли, потому что первым шагом мы уже переместили все единицы в свой блок.
- Блок троек сортировки не требует, потому что на предыдущих шагах мы уже переместили единицы в блок единиц и двойки в блок двоек.

Тест производительности



По результатам тестирования видно, что сложность выполнения алгоритма является линейной. Потому что за $O(n)$ мы считываем последовательность и одновременно подсчитываем количество единиц, двоек и троек. Далее еще за $O(n)$, пройдясь по массиву, мы посчитаем количество элементов, находящихся не в своих блоках. И сами обмены будут выполнены не более чем за $O(n)$, потому что в каждой итерации цикла мы уменьшаем количество неотсортированных элементов на 1, а всего таких элементов не может быть больше, чем n . Тогда $O(n + n + n) = O(3n) = O(n)$. Решение этой задачи жадным алгоритмом соответствует сложности $O(n)$, где n — длина входной последовательности.

Выводы

В результате проведенной лабораторной работы я познакомилась с жадными алгоритмами и реализовала один из них. В жадном алгоритме мы на каждом шаге делаем локально наилучший выбор в надежде на то, что итоговое решение будет оптимальным, что отличается от динамического программирования. Решение о применении жадного алгоритма должно приниматься заранее, поскольку этот метод не всегда дает оптимальный ответ.