Лабораторная работа № 8 по курсу дискретного анализа: Жадные алгоритмы

Выполнил студент группы М8О-308Б-20 МАИ *Зинин Владислав*.

Условие

1. Разрабтать жадный алгоритм решения задачи, определяемой своим вариантом. Доказать его корректность, оценить скорость и объём затрачиваемой оперативной памяти.

2. **Вариант 5: Оптимальная сортировка чисел.** Дана последовательность длины N из целых чисел 1, 2, 3. Необходимо найти минимальное количество обменов элементов последовательности, в результате которых последовательность стала бы отсортированной.

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. Предположим, что у нас есть 3 границы для чисел $i = 1, 2, 3$ соответственно, в которых должны находится только числа i, чтобы массив был отсортирован. Для определения границ зон при считывании данных нам надо подсчитать количество вхождения каждого числа. Обходя массив слева направо, если мы обнаруживаем элемент, который находится не в своей зоне, мы будем сразу же заменять его на нужный элемент, взятый из другой, подходящей для данного случая зоны. Таким образом, может возникнуть 3 ситуации: 1. Мы обнаружили в зоне 1 число 2, значит нужно найти во второй (а если во второй нет, то в третьей) зоне число 1 и поменять числа местами. 2. Мы обнаружили в зоне 1 число 3, значит нужно найти в третьей (е если в третьей нет, то во второй) зоне число 1 и поменять числа местами. 3. Мы обнаружили в зоне 2 число 3, значит нужно найти в третьей зоне число 2 и поменять числа местами. Для определения границ между зонами создадим массив $count$, в котором будет вестись подсчат количества каждого числа. $count[i]$ - количество чисел $i + 1$ в исходных данных.

Таким образом, мы определяем границы зон следующим образом: 1. $[0; count[0]]$ - зона 1 2. $[count[0]; count[0] + count[1]]$ - зона 2 3. $[count[0] + count[1]; N]$ - зона 3, N - длина массива. Введем три переменные - $pos3_1$, $pos2_1$, $pos3_2$ где $posi_j$ - позиция числа j в зоне i. В первом и третьем случае мы всегда берем самое левое подходящее число. Во втором случае - самое правое. Это нужно, чтобы все 3 оказались в правой части массива. Для обновления данных переменные реализована функция $NextPosition$.

Описание программы

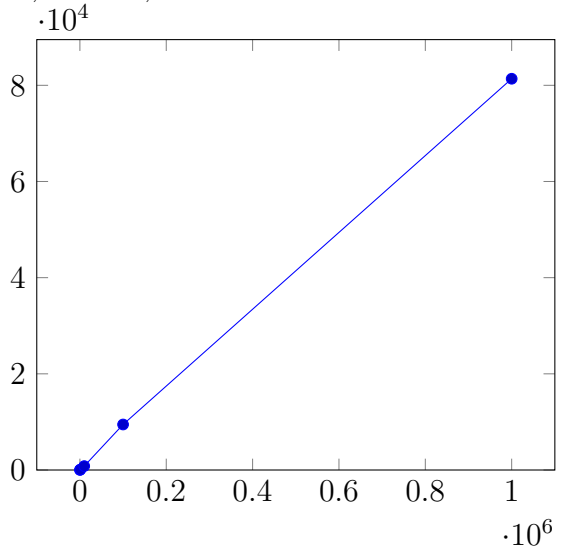
Программа состоит из одного файла.

Дневник отладки

1. Мелкие ошибки в *swap* функциях привели к долгому выявлению их, из-за чего время выполнения данной лабораторной сильно увеличилось.

Тест производительности

Убедимся, что данный алгоритм действительно имеет линейную сложность. Для этого замерим время работы программы для последовательностей следующих длин: 100, 1000, 10000, 100000, 1000000.



По графику мы видим, что сложность действительно $O(n)$.

Выводы

В результате проведенной лабораторной работы я познакомился с жадными алгоритмами и реализовал один из них. Как я упомянул ранее, 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, поскольку во втором случае мы принимаем решение на каждом шаге с учетом текущей проблемы и решения ранее решенной подзадачи для нахождения оптимального решения. Решение о применении жадного алгоритма должно приниматься заранее, поскольку этот метод не всегда дает оптимальное решение.