Санкт-Петербургский Национальный Исследовательский Университет Информационных Технологий, Механики и Оптики

Факультет инфокоммуникационных технологий

**Лабораторная работа №3**

Выполнили:

Бабаев Р.С.

Садовая А.Р.

Абдулов И.А.

Проверил:

Мусаев А.А.

Санкт-Петербург,

2022

Оглавление

[Введение 3](#_Toc118406971)

[1 Задание №1 4](#_Toc118406972)

[2 Задание №2 5](#_Toc118406973)

[3 Задание №3 7](#_Toc118406974)

[Заключение 8](#_Toc118406975)

[Список литературы 9](#_Toc118406976)

# **Введение**

Данная работа представляет собой отчет о выполненных заданиях:

1. Написать программу для сортировки массива пузырьком и сравнить ее эффективность с методом sort().
2. Придумать алгоритмы, имеющие сложность О(3n), О(nlog(n)), О(n!), О(n^3), О(3log(n))
3. Построить зависимость между количеством элементов и количеством шагов для алгоритмов со сложностями О(1), О(log(n)), О(n^2), О(2^n). Сравнить сложности алгоритмов.

# **Задание №1**

Сортировка пузырьком заключается в том, что элемент сравнивается со следующим и, если он оказывается больше, то они меняются местами. Таким образом каждый элемент сравнивается с элементами, которые стоят «выше» него, и останавливается, как только следующий встречающийся элемент оказывается больше или равен текущему. Также можно пользоваться просто методом sort()

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1.1 – Код программы для сортировки пузырьком и через метод sort()



Рисунок 1.2 – Пример работы программы

Как мы видим, результат получен одинаковый. Посмотрим, чем тогда отличаются два этих способа.

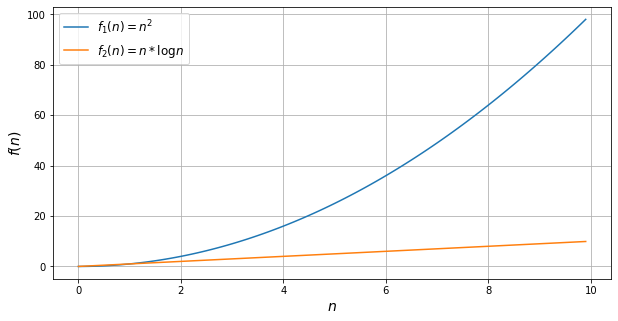


Рисунок 1.3 – Сравнение эффективности двух способов сортировки

Вывод:

Сортировка пузырьком имеет большую сложность, чем сортировка через метод sort(). Кроме того, чем больше элементов в массиве, тем больше различие в сложности исполняемого алгоритма. Однако оба этих способа подходят для сортировки массива, просто один более оптимизированный.

# **Задание №2**

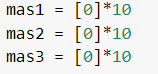


Рисунок 2.1 – Алгоритм со сложностью О(3n) – создание трёх пустых массивов

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.2 – Алгоритм со сложностью O(n(log(n))) – поиск индексов элементов массива в отсортированном массиве

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.3 – Алгоритм со сложностью O(n!) – количество перестановок

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.4 – Алгоритм со сложностью O(n3) – 3 цикла

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.5 – Алгоритм со сложностью O(3(log(n))) – 3 бинарных поиска

# **Задание №3**

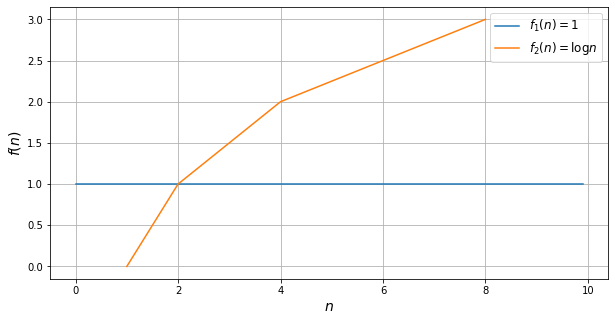


Рисунок 3.1 – Зависимость между количеством элементов и количеством шагов для алгоритмов сложности О(1) и О(log(n))d

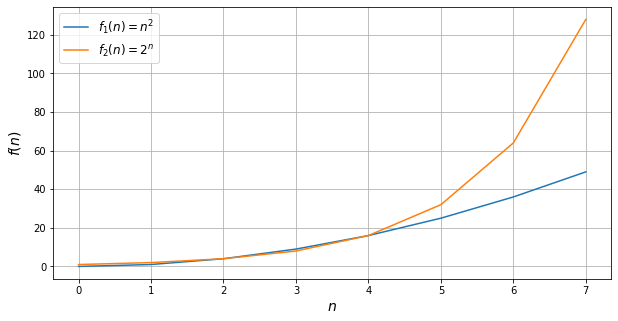


Рисунок 3.2 – Зависимость между количеством элементов и количеством шагов для алгоритмов сложности от количества шагов О(n2) и О(2n)

# **Заключение**

Таким образом, для каждой задачи была написана программа на языке программирования Python. Был изучен алгоритм сортировки массива пузырьком и рассмотрен встроенный метод sort(). Также написаны алгоритмы с различными сложностями. Для каждого решения приведены программы.

Все программы можно найти на GitHub [1].

# **Список литературы**

1. GitHub [Электронный ресурс] – <https://github.com/estle/itmo-uni/tree/main/sem1/ADS/lab3> (Дата обращения – 03.11.2022).