Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

Кафедра Систем Управления и Информатики

**Лабораторная работа №4**

Выполнили:

Пархомов С.А.

Проверил:

Мусаев А.А.

Санкт-Петербург,

2022

Задание №1

При выполнении данного задания мы ознакомились с двумя методами сортировки: Сортировка расческой и Быстрая сортировка, а также с использованием модулей в Python. Замеры времени исполнения программы, сделанные посредством использования библиотеки Timeit для 1000 случайных чисел в пределах от -1000 до 1000, показали, что алгоритм для Быстрой сортировки (~0,005с) работает во много раз быстрее, чем алгоритм для сортировки расческой (~0,346с).

Код программы представлен в файле: Лаба4 Задание1.ру

Код модуля представлен в файле: Module1.ру

Ссылка на GitHub: [SergeyParkhomov/Laba\_04 (github.com)](https://github.com/SergeyParkhomov/Laba_04)

Задание №2

В ходе выполнения данного задания мы ознакомились с пирамидальной и блочной сортировками, а также затронули сортировку вставки в алгоритме блочной сортировки для упрощения процесса, не меняя принцип – в блочной сортировке при сортировке внутри блоков можно использовать любой другой метод сортировки (в нашем случае - сортировка вставками), либо рекурсивно этот же).

Код программы представлен в файле: Лаба4 Задание2.ру

Ссылка на GitHub: [SergeyParkhomov/Laba\_04 (github.com)](https://github.com/SergeyParkhomov/Laba_04)

Задание №3

Блочная сортировка имеет сложность О(n) при «удачных входных данных», однако она быстро деградирует при большом количестве мало отличных элементов.

Сортировка вставками имеет сложность О(n2).

Пирамидальная сортировка имеет сложность О(nlog2n).

Быстрая сортировка имеет сложность О(nlogn), но при «неудачных входных» способна сильно деградировать по сложности до О(n2).

Сортировка расческой имеет сложность О(n2), однако стремится к О(nlogn).

Замеры времени при помощи TimeIt показали, что Быстрая сортировка в нашем случае эффективнее ( и по моему личному мнению намного понятнее и удобнее), однако для разных типов входных данных, их количества и различных целей надо подбирать методом сортировки индивидуально, так как некоторые из них могут быть неустойчивы в тех или иных случаях или «съедать больше памяти» - так что всё зависит от целей программы и её возможностей.