Министерство науки и образования РФ

Федеральное государственное бюджетное учреждение

высшего образования

**«Тверской государственный технический университет»**

(ТвГТУ)

Кафедра программного обеспечения

**Отчет по лабораторной работе №1**

по дисциплине: «Теория алгоритмов»

Тема: «Введение в разработку консольных приложений на python»

|  |
| --- |
| Выполнил:  студент группы  Б.ПИН.РИС-21.06  Леонов А.М. |
| Проверила:  старший преподаватель кафедры ПО  Корнеева Е.И. |

Тверь 2023

**Задачи**Написать алгоритм Шейкер сортировки (Shaker sort) на языке Python. Реализовать вход данных в программу из файла и выход результата работы программы в файл. Сравнить работу алгоритма на различных объёмах входных данных и построить графики.

**Алгоритм решения**

1. Реализовать функцию сортировки через def.

2. Реализовать ввод данных из файла .txt

3. Реализовать вывод данных в файл .txt

4. Автоматизировать тестирование через for

**Алгоритм сортировки:**

1. Получаем длину массива и устанавливаем флаг swapped в True, start\_index в 0, end\_index в длину массива - 1.

2. Запускаем цикл while, пока swapped равен True.

3. Устанавливаем флаг swapped в False.

4. Запускаем цикл for для прохода слева на право от start\_index до end\_index.

5. Если текущий элемент больше следующего, то меняем их местами и устанавливаем флаг swapped в True.

6. Если swapped равен False, то прерываем цикл while.

7. Устанавливаем флаг swapped в False и уменьшаем end\_index на 1.

8. Запускаем цикл for для прохода справа налево от end\_index-1 до start\_index-1.

9. Если текущий элемент больше следующего, то меняем их местами и устанавливаем флаг swapped в True.

10. Увеличиваем start\_index на 1.

11. Повторяем шаги 3-10, пока swapped равен True.

12. Возвращаем отсортированный массив.

**Тестирования программы:**Тестирование было выполнено на различных объемах входных данных: текстовые файлы с числами от 0 до 100 по 10, 100, 1000, 10000,100000 чисел в каждом файле (Рис.1).

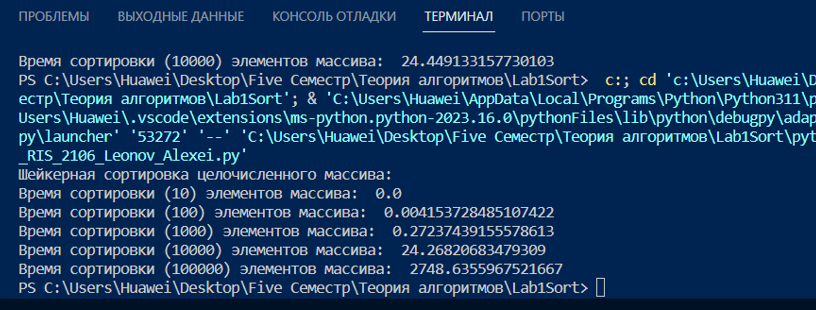
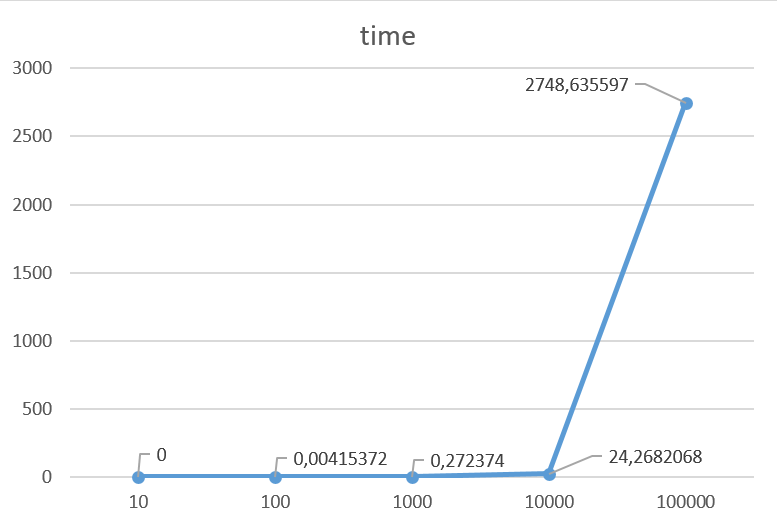
  
Рис.1 Время сортировки (n) элементов массива.

График зависимости времени сортировки от количества входных данных:

  
Рис.2 График.

На графике (Рис.2) мы можем увидеть, что с увеличением объёма данных которые нам нужно отсортировать время работы программы неумолимо растёт.

**Ссылка на репозиторий с программой:**<https://github.com/akkafe1ix/Lab1Sort>

**Вывод:**Алгоритм Шейкер сортировки является усовершенствованным вариантом пузырьковой сортировки. Она позволяет более эффективно удалять элементы, которые уже находятся на своих местах. В результате получается отсортированный массив. Однако, Шейкер сортировка не является самым быстрым алгоритмом сортировки и не рекомендуется использовать для больших массивов. Написанная программа наглядно показывает время работы Шейкер сортировки на различных объёмах входных данных, и позволяет объективно оценить её работоспособность.