

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
По практической работе №5
Дисциплины «Алгоритмизация»

Выполнил:

Пустяков Андрей Сергеевич

2 курс, группа ИВТ-б-о-22-1,

09.03.01 «Информатика и
вычислительная техника (профиль)
«Программное обеспечение средств
вычислительной
техники и автоматизированных
систем», очная форма обучения

(подпись)

Руководитель практики:

Воронкин Р. А. кандидат технических
наук, доцент, доцент кафедры
инфокоммуникаций

(подпись)

Ставрополь, 2023 г.

Тема: Анализ времени работы алгоритма сортировки методом пузырька.

Цель: проанализировать время работы алгоритма сортировки методом пузырька, используя метод наименьших квадратов построить график зависимости времени работы алгоритма от размера списка.

Ход работы:

Необходимо создать программу, которая создает данные для анализа скорости работы алгоритма сортировки списка методом пузырька.

На основе полученных данных о затраченном времени на сортировку элементов в списках различной длины необходимо построить точечный график. С использованием метода наименьших квадратов необходимо построить график функции.

Код программы данного задания на языке программирования Python и результаты работы программы (данные о времени сортировке в файле «results.txt») (рис. 1, 2).

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import timeit
import random

def bubble_sort(list_to_sort):
    """
    Функция сортировки массива методом пузырька.
    """
    n = len(list_to_sort)
    for i in range(n):
        for j in range(0, n - i - 1):
            if list_to_sort[j] > list_to_sort[j + 1]:
                list_to_sort[j], list_to_sort[j + 1] = list_to_sort[j + 1],
list_to_sort[j]
        return None

if __name__ == '__main__':
    # Необходимо создать программу, которая создает данные для анализа
    скорости работы
    # алгоритма сортировки списка методом пузырька.

    current_list = [random.randint(-100, 100) for i in range(1000)]

    # Для записи результатов используется файл с данными в той же директории,
    что и данный модуль.
    filename_results = 'results.txt'
```

```

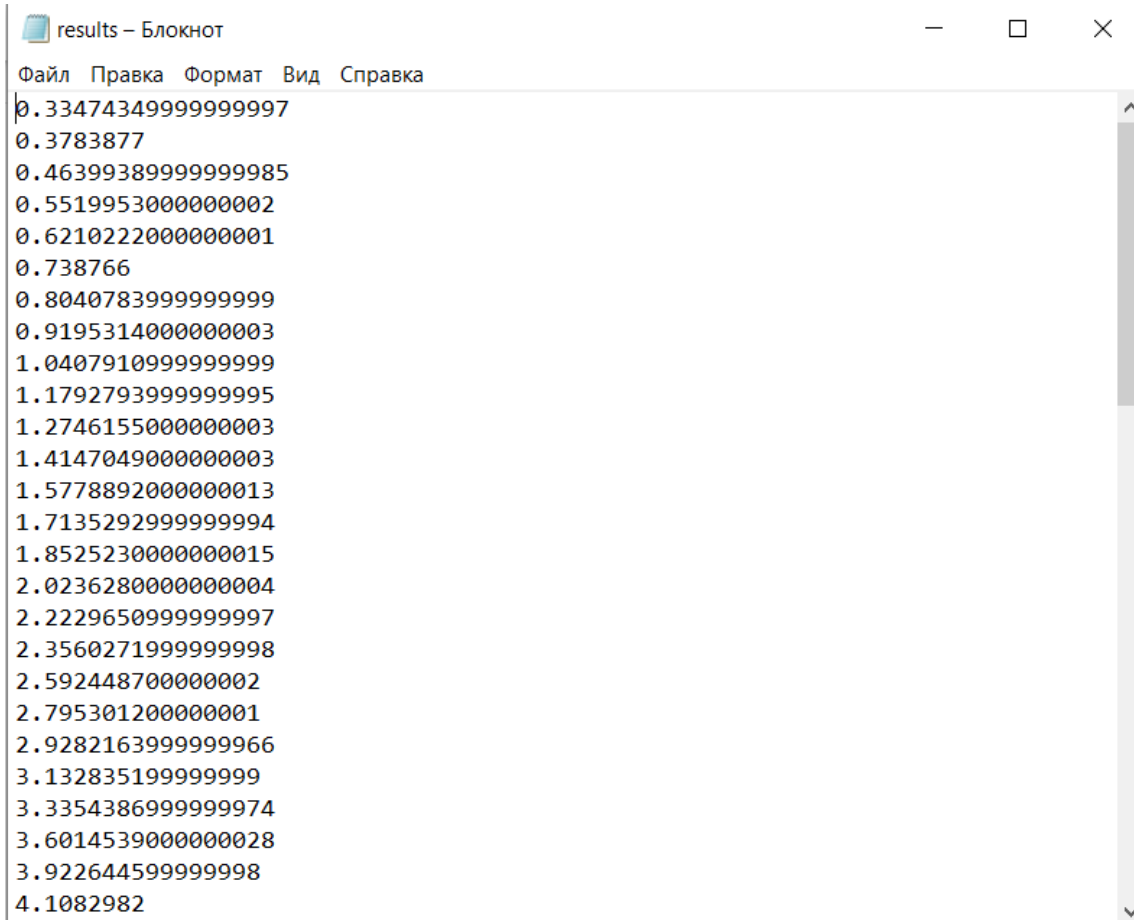
# Получим данные времени работы алгоритма.
for number in range(0, 70):

    with open(filename_results, 'a') as file:
        time_search = timeit.timeit(lambda: bubble_sort(current_list),
number=10)
        file.write(str(time_search) + '\n')
    file.close()

# Увеличим размер данного списка.
for i in range(0, 100):
    current_list.append(random.randint(-100, 100))

```

Рисунок 1 – Код программы получения данных о сортировке списков.



```

results - Блокнот
Файл  Правка  Формат  Вид  Справка
0.3347434999999997
0.3783877
0.46399389999999985
0.5519953000000002
0.6210222000000001
0.738766
0.8040783999999999
0.9195314000000003
1.0407910999999999
1.1792793999999995
1.2746155000000003
1.4147049000000003
1.5778892000000013
1.7135292999999994
1.8525230000000015
2.0236280000000004
2.2229650999999997
2.3560271999999998
2.5924487000000002
2.7953012000000001
2.9282163999999966
3.1328351999999999
3.3354386999999974
3.6014539000000028
3.9226445999999998
4.1082982

```

Рисунок 2 – Содержимое файла «results.txt».

График зависимости времени сортировки списка методом пузырька от размера списка (рис. 4).

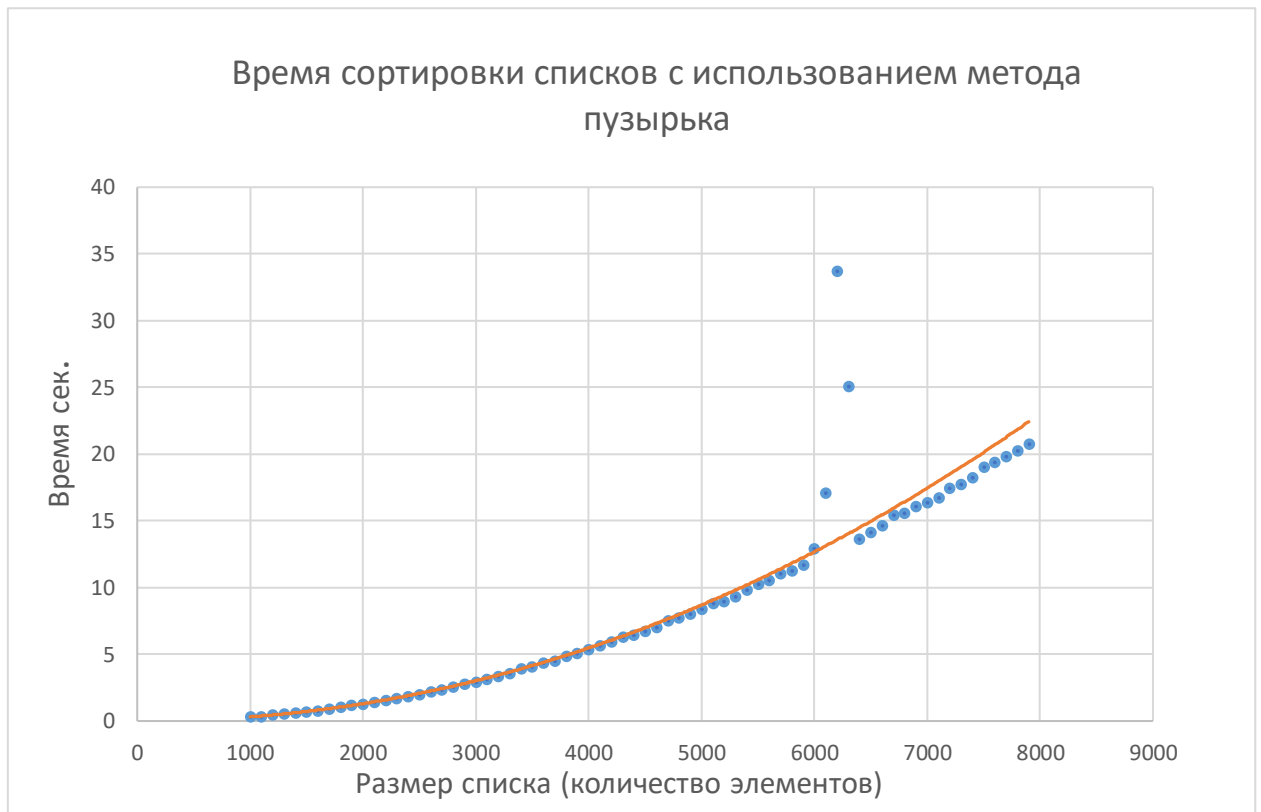


Рисунок 4 – Точечный график зависимости времени поиска максимального элемента в списке от размера списка.

Вывод: в ходе выполнения лабораторной работы были рассмотрен алгоритм сортировки списка методом пузырька. Был проведен анализ времени работы алгоритма. В зависимости от размеров списка со случайными элементами время растет согласно степенной функции. Был построен график зависимости времени сортировки от размеров списка. Исходя из такой скорости роста можно сделать вывод о том, что алгоритм сортировки списка методом пузырька является довольно медленным при относительно больших размерах списков.