

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 №4**  
**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: Алгоритмы линейного поиска. Поиск максимального и минимального элемента в списке

Цель: проанализировать скорость нахождения минимального и максимального элемента в списке с различным размером списка, построить график используя метод наименьших квадратов.

Ход работы:

Необходимо создать программу, которая создает данные для анализа скорости работы линейного поиска в списке минимального элемента и максимального элемента.

На основе полученных данных о затраченном времени на поиск 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 поиска в двух разных текстовых файлах) (рис. 1, 2, 3). Полученные данные помещаются в два отдельных файла (для результатов поиска минимального и максимального элемента в списке «min\_results.txt» и «max\_results.txt»).

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

import timeit
import random

def max_search(list_search):
    """
    Функция поиска максимального элемента в переданном ей списке.
    """
    max_element = list_search[0]
    for i in list_search:
        if max_element < i:
            max_element = i
    return max_element

def min_search(list_search):
    """
    Функция поиска минимального элемента в переданном ей списке.
```

```

"""
min_element = list_search[0]
for i in list_search:
    if i < min_element:
        min_element = i
return min_element

if __name__ == '__main__':
    # Необходимо создать программу, которая создает данные для анализа
    скорости работы
    # линейного поиска в списке минимального элемента и максимального
    элемента.

    current_list = [random.randint(-100, 100) for i in range(100000)]

    # Для записи результатов используются два файла в той же директории, что
    и данный модуль.
    filename_min = 'min_results.txt'
    filename_max = 'max_results.txt'

    # Получим данные при поиске максимального элемента в списке.
    for number in range(0, 400):

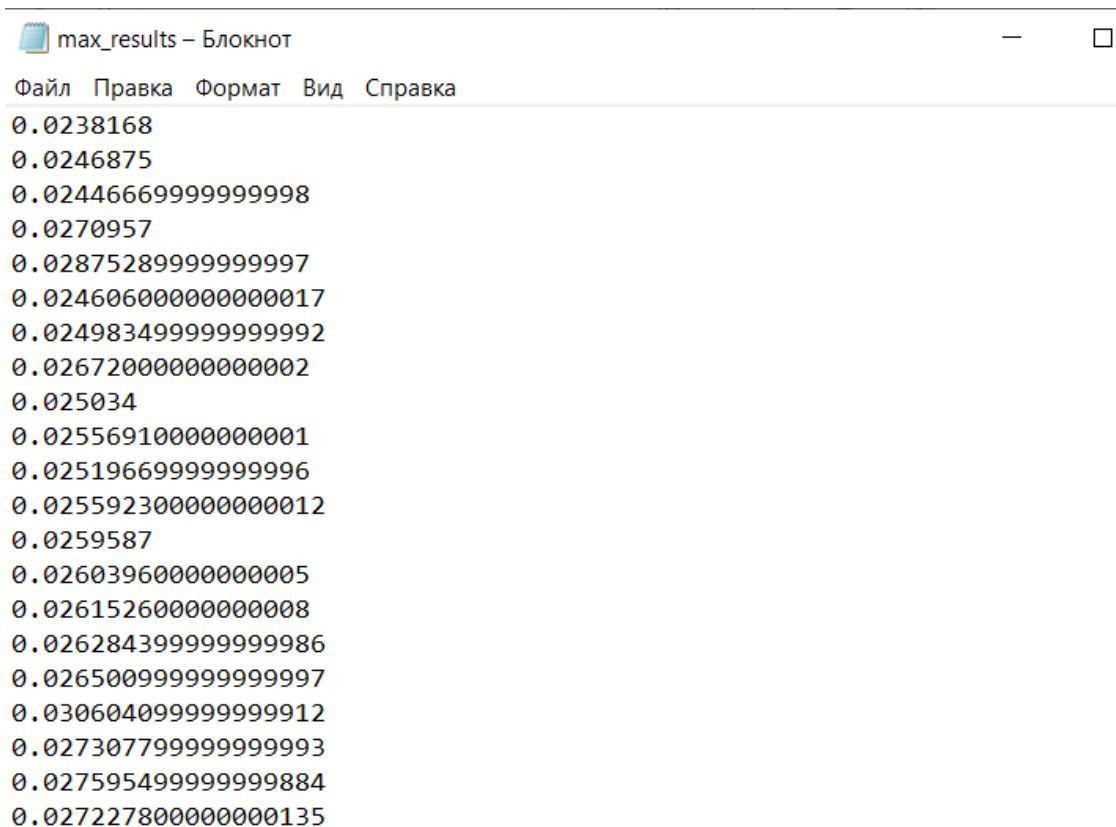
        with open(filename_max, 'a') as file:
            time_search = timeit.timeit(lambda: max_search(current_list),
number=10)
            file.write(str(time_search) + '\n')
            file.close()

        # Получим данные при поиске минимального элемента в списке.
        with open(filename_min, 'a') as file:
            time_search = timeit.timeit(lambda: min_search(current_list),
number=10)
            file.write(str(time_search) + '\n')
            file.close()

        # Увеличим размер данного списка.
        for i in range(1, 1001):
            current_list.append(random.randint(-100, 100))

```

Рисунок 1 – Код программы получения данных о времени поиска.

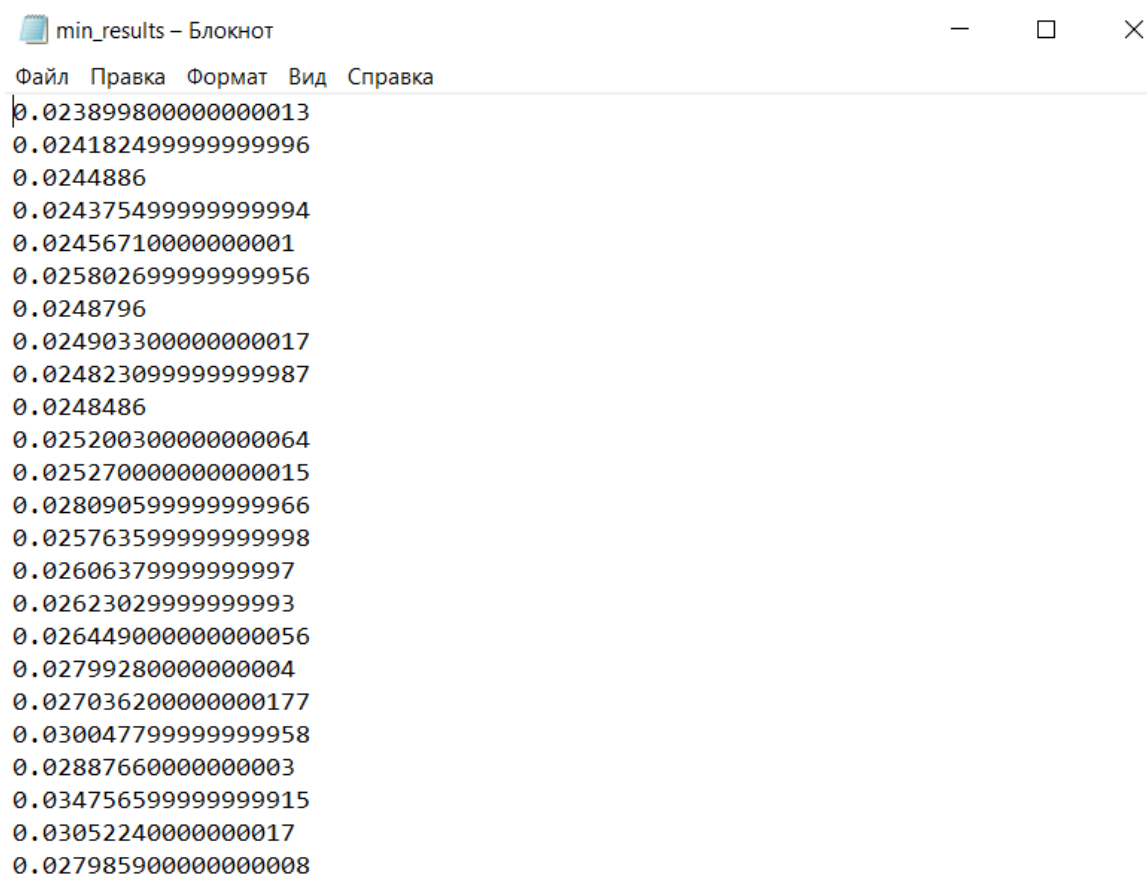


max\_results – Блокнот

Файл Правка Формат Вид Справка

```
0.0238168
0.0246875
0.02446669999999998
0.0270957
0.02875289999999997
0.024606000000000017
0.024983499999999992
0.026720000000000002
0.025034
0.025569100000000001
0.02519669999999996
0.025592300000000012
0.0259587
0.026039600000000005
0.026152600000000008
0.026284399999999986
0.02650099999999997
0.030604099999999912
0.02730779999999993
0.027595499999999884
0.027227800000000135
```

Рисунок 2 – Содержимое файла «max\_results.txt».



min\_results – Блокнот

Файл Правка Формат Вид Справка

```
0.0238998000000000013
0.024182499999999996
0.0244886
0.024375499999999994
0.024567100000000001
0.025802699999999956
0.0248796
0.0249033000000000017
0.024823099999999987
0.0248486
0.0252003000000000064
0.0252700000000000015
0.028090599999999966
0.025763599999999998
0.02606379999999997
0.02623029999999993
0.0264490000000000056
0.027992800000000004
0.02703620000000000177
0.030047799999999958
0.028876600000000003
0.034756599999999915
0.0305224000000000017
0.0279859000000000008
```

Рисунок 3 – Содержимое файла «min\_results.txt».

График зависимости времени нахождения максимального элемента в списке от размера списка (рис. 4). Коэффициент Пирсона (парной корреляции) в данном случае = 0,992660801 (близок к единице).



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

График зависимости времени нахождения минимального элемента в списке от размера списка (рис. 5). Коэффициент Пирсона (парной корреляции) в данном случае = 0,992588251 (также близок к единице).

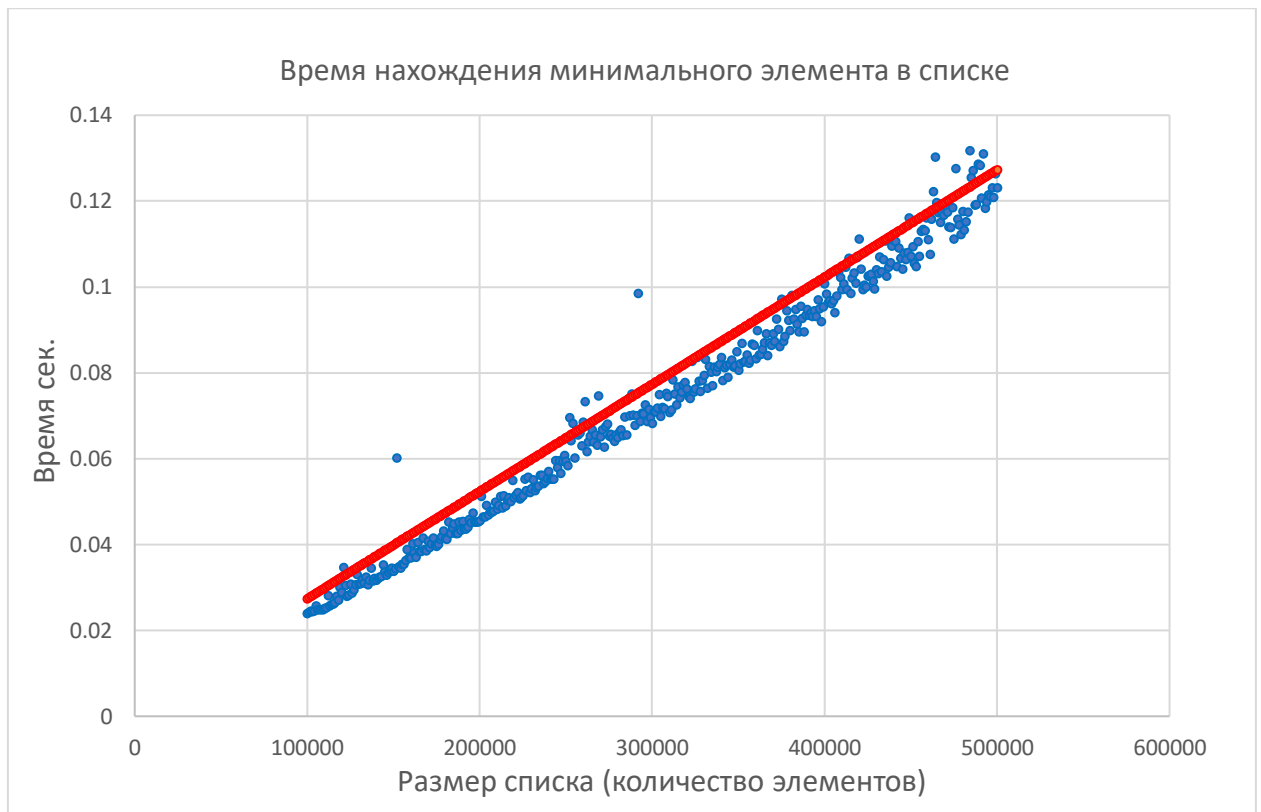


Рисунок 5 – Точечный график зависимости времени поиска минимального элемента от размера списка.

Вывод: в ходе выполнения лабораторной работы были рассмотрены алгоритмы поиска максимального и минимального элементов в списке. Был проведен анализ времени работы таких алгоритмов. В зависимости от размеров списка со случайными элементами время растет линейно. Были построены графики линейной зависимости времени работы от размеров списка. Полученные коэффициенты парной корреляции говорят о тесной взаимосвязи одной переменной от другой (близки к единице).