

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 ЛАБОРАТОРНОЙ РАБОТЕ №4  
дисциплины «Алгоритмизация»**

Выполнил:  
Епифанов Алексей Александрович  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной  
техники и автоматизированных систем  
», очная форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Роман Александрович

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2023 г.

### Порядок выполнения работы:

1. Написал программу, которая выводит в виде графиков зависимость времени выполнения функций поиска минимума и максимума от размера массива. Также на графике есть прямая, построенная по методу меньших квадратов, а в консоль выводится коэффициент парной корреляции.

```
main.py X
algorithm4 > program > main.py > ...
1 import random as rnd
2 import matplotlib.pyplot as plt
3 import numpy as np
4 import timeit
5
6
7 def findmin():
8     min = randmax
9     for i in a:
10         if min > i:
11             min = i
12     return min
13
14
15 def findmax():
16     max = 0
17     for i in a:
18         if max < i:
19             max = i
20     return max
21
22
23 def create_graph(b, c, aur, bur, namegraph):
24     plt.scatter(b, c, s=5)
25     y_line = aur * np.array(b) + bur
26     plt.plot(b, y_line, color='red')
27     plt.title(namegraph)
28     plt.xlabel("Размер массива")
29     plt.ylabel("Время работы функции")
30     correlation_coefficient = np.corrcoef(c, b)[0, 1]
31     return correlation_coefficient
32
33
34 correlation_v = []
35 # Цикл нужен для создания двух графиков, один при минимуме, второй при максимуме
36 for namegraph in ["Минимум", "Максимум"]:
37     x = [i for i in range(10, 10001, 10)]
38     time = []
39     x2 = []
40     xtime = []
41     randmax = 1000000
42     if namegraph == "Минимум":
43         for i in x:
44             a = [rnd.randint(0, randmax) for j in range(i)]
45             time.append((timeit.timeit(lambda: findmin(), number=50))/50)
46     else:
47         for i in x:
48             a = [rnd.randint(0, randmax) for j in range(i)]
49             time.append((timeit.timeit(lambda: findmax(), number=50))/50)
50
51     # Вычисление коэффициентов в системе уравнений метода наименьших квадратов
52     sx = sum(x)
53     stime = sum(time)
54     sx2 = sum(i**2 for i in x)
55     sxtime = sum(i*j for i, j in zip(x, time))
56     n = len(x)
57     # k - это коэффициент, при котором вычитание
58     # из первого уравнения второго,
59     # умноженного на него, приводит к нулю в коэффициенте при x.
60     # Таким образом, мы сможем вычислить свободный коэффициент.
61     k = sx2/sx
62     # bur - это свободный коэффициент
63     bur = (sxtime - k*stime)/(sx-k*n)
64     # aur - это коэффициент при x
65     aur = (stime - bur*n)/sx
66     # Создание графических окон
67     plt.figure(namegraph)
68     plt.subplots_adjust(left=0.2)
69
70     # Создание графиков
71     correlation_v.append(create_graph(x, time, aur, bur, namegraph))
72
73 print("Коэффициент корреляции в первом случае =",
74       correlation_v[0], "\на во втором случае =", correlation_v[1])
75
76 # Показ графиков
77 plt.show()
78
```

Рисунок 1. Код программы

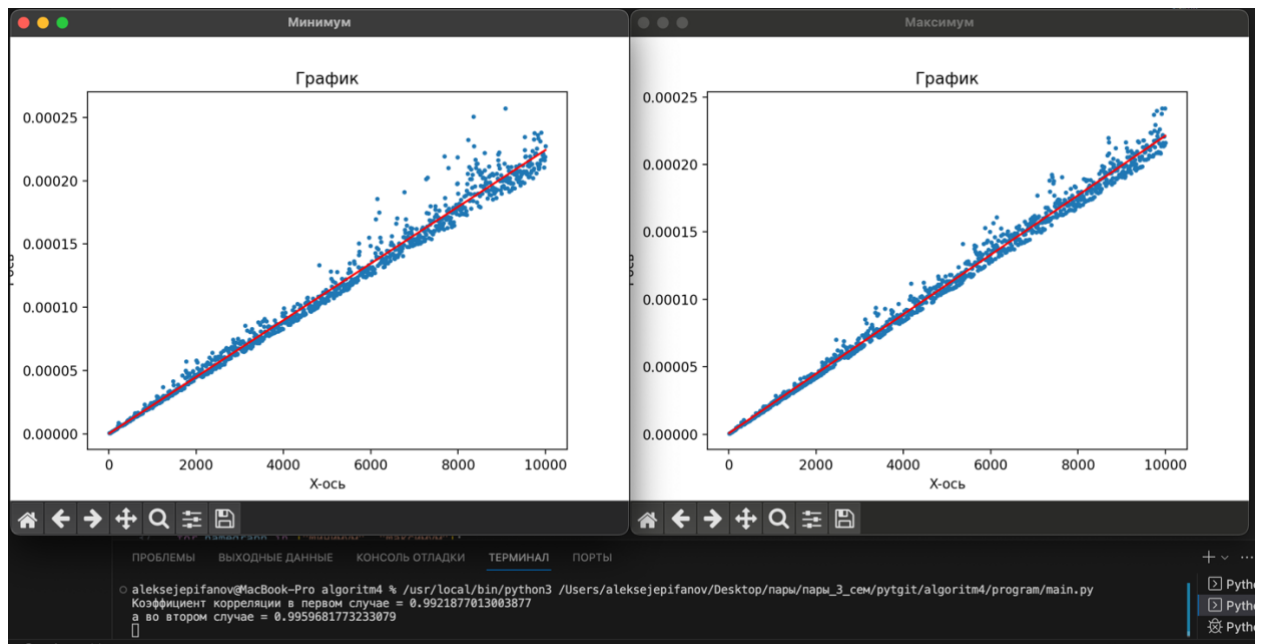


Рисунок 2. Вывод графиков и вывод в консоли

В ходе выполнения лабораторной работы был проведен анализ зависимости времени выполнения функций поиска минимума и максимума от размера массива. Из полученных результатов можно сделать следующий вывод: время работы функций поиска минимума и максимума линейно зависит от размера массива.