

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:  
Магдаев Даламбек Магомедович  
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 Р.А., доцент кафедры  
инфокоммуникаций

---

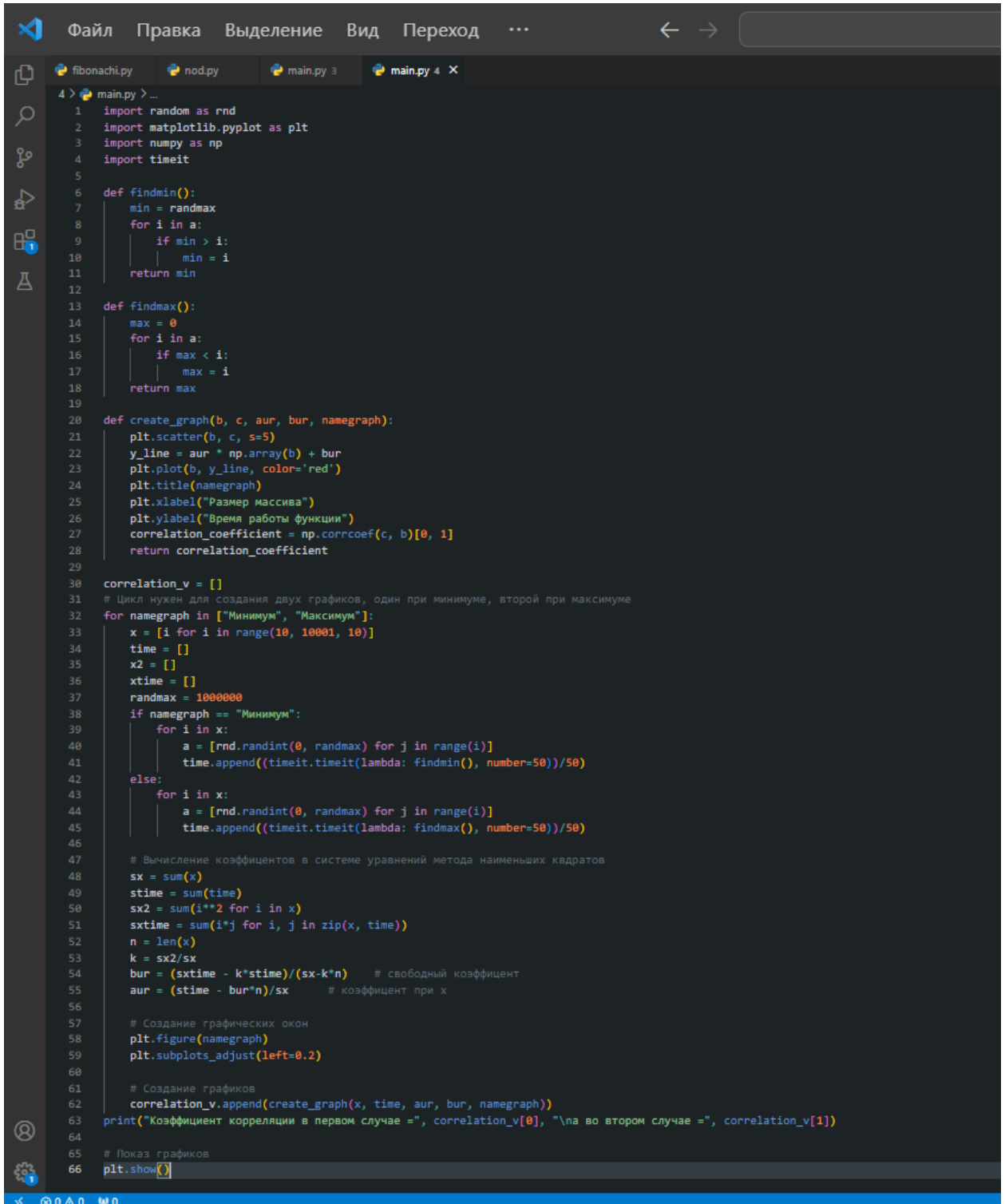
(подпись)

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.

A screenshot of a code editor with a dark theme. The editor has a menu bar at the top with 'Файл', 'Правка', 'Выделение', 'Вид', 'Переход', and '...'. Below the menu bar is a tab bar showing four tabs: 'fibonachi.py', 'nod.py', 'main.py 3', and 'main.py 4'. The active tab is 'main.py 4'. The code is written in Python and includes imports for random, matplotlib.pyplot, numpy, and timeit. It defines functions findmin(), findmax(), and create\_graph(). The main part of the code is a loop that generates data for finding min and max, calculates the correlation coefficient, and plots the results. The code ends with a call to plt.show().

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

Рисунок 1. Код программы

## 2) Результат выполнения программы:

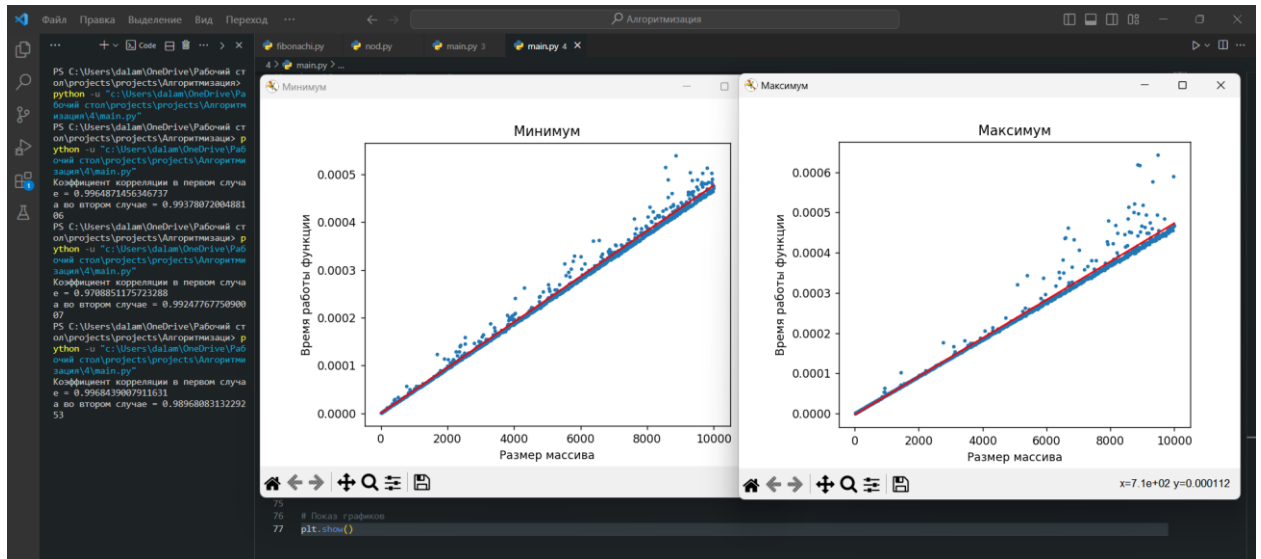


Рисунок 2. Вывод графиков

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.