

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:  
Середа Кирилл Витальевич  
1 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника», очная  
форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Роман Александрович

---

(подпись)

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

Ставрополь, 2023 г.

Тема: Анализ Алгоритмов. Алгоритм поиска максимума и минимума

Цель: Произвести анализ алгоритма поиска максимума и минимума, вычислить коэффициент парной корреляции и составить линейную зависимость.

Ход выполнения заданий

1) Написал программу для аналитики алгоритмов поиска максимума и минимума.

Код программы:

```
import timeit
import random
import matplotlib.pyplot as plt
import numpy as np

# region Функции

def MinSearch_experiment(arr_size):
    arr = random.sample(range(1, arr_size + 1), arr_size)
    execution_time = timeit.timeit(lambda: min(arr), number=1000)
    return execution_time

def MaxSearch_experiment(arr_size):
    arr = random.sample(range(1, arr_size + 1), arr_size)
    execution_time = timeit.timeit(lambda: max(arr), number=1000)
    return execution_time

# endregion

# region Параметры
array_sizes = [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000,
               2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000]
num_experiments = 5
min_execution_times = []
max_execution_times = []
# endregion

for size in array_sizes:
    min_times = []
    max_times = []

    for _ in range(num_experiments):
        min_time = MinSearch_experiment(size)
        max_time = MaxSearch_experiment(size)

        min_times.append(min_time)
        max_times.append(max_time)

    # Подсчет среднего времени
    avg_min_time = sum(min_times) / num_experiments
    avg_max_time = sum(max_times) / num_experiments

    min_execution_times.append(avg_min_time)
    max_execution_times.append(avg_max_time)

# region Выполнение линейной регрессии и расчет коэффициента корреляции для
```

```

обоих случаев.
x = np.array(array_sizes)
y_min = np.array(min_execution_times)
y_max = np.array(max_execution_times)

A_min = np.vstack([x, np.ones(len(x))]).T
a_min, b_min = np.linalg.lstsq(A_min, y_min, rcond=None)[0]

A_max = np.vstack([x, np.ones(len(x))]).T
a_max, b_max = np.linalg.lstsq(A_max, y_max, rcond=None)[0]

correlation_coefficient_min = np.corrcoef(x, y_min)[0, 1] ** 2
correlation_coefficient_max = np.corrcoef(x, y_max)[0, 1] ** 2
# endregion

# region Визуализация
plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
plt.plot(x, y_min, 'o', label='Поиск минимума')
plt.plot(x, a_min * x + b_min, 'r', label=f'Коэффициент корреляции (R^2={correlation_coefficient_min:.5f})')
plt.xlabel('Размер массива')
plt.ylabel('Время выполнения (секунды)')
plt.title('Аналитика поиска минимума')
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(x, y_max, 'o', label='Поиск максимума')
plt.plot(x, a_max * x + b_max, 'g', label=f'Коэффициент корреляции (R^2={correlation_coefficient_max:.5f})')
plt.xlabel('Размер массива')
plt.ylabel('Время выполнения (секунды)')
plt.title('Аналитика поиска максимума')
plt.legend()

plt.tight_layout()
plt.show()
# endregion

# region Вывод в консоль
print(f"Поиск минимума: Линейная зависимость:  $y = \{a\_min:.5f\} * x + \{b\_min:.5f\}$ ")
print(f"Поиск минимума: Коэффициент линейной корреляции: {correlation_coefficient_min:.5f}")

print(f"Поиск максимума: Линейная зависимость:  $y = \{a\_max:.5f\} * x + \{b\_max:.5f\}$ ")
print(f"Поиск максимума: Коэффициент линейной корреляции: {correlation_coefficient_max:.5f}")
# endregion

```

Полученные результаты:

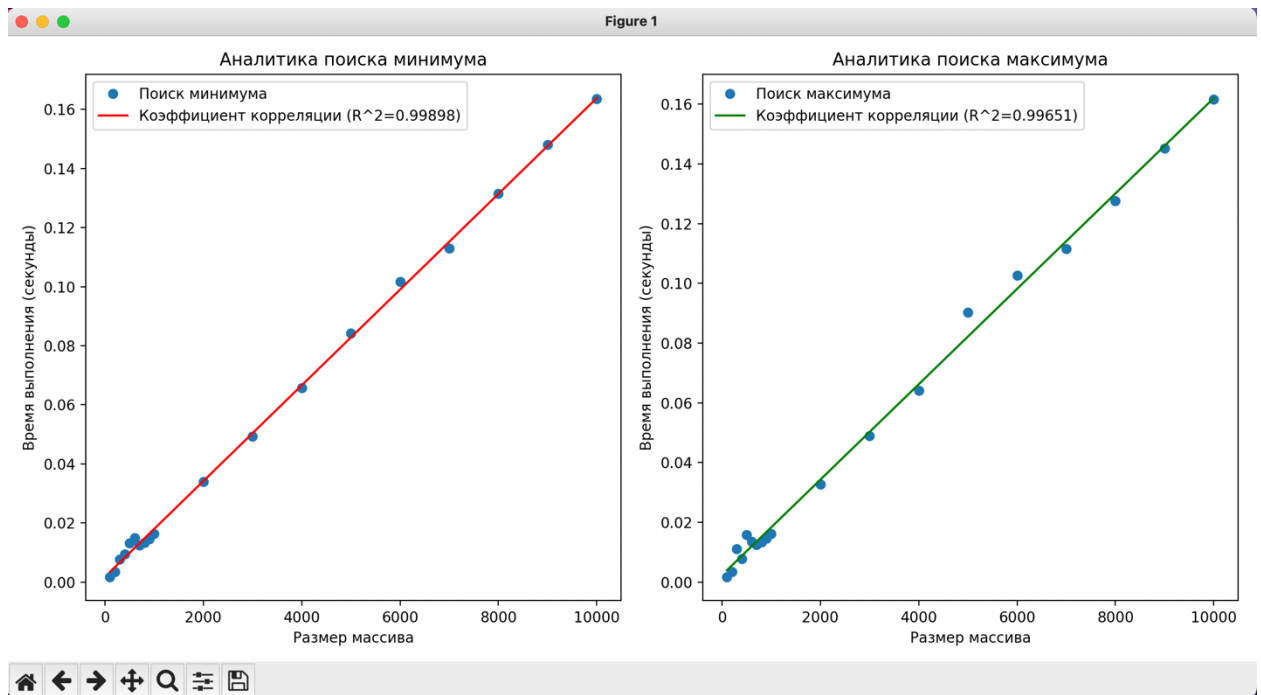


Рисунок 1 – Графики с аналитикой

```
Поиск минимума: Линейная зависимость:  $y = 0.00002 * x + 0.00179$   
Поиск минимума: Коэффициент линейной корреляции: 0.99898  
Поиск максимума: Линейная зависимость:  $y = 0.00002 * x + 0.00234$   
Поиск максимума: Коэффициент линейной корреляции: 0.99651
```

Рисунок 2 – Полученные данные

Вывод: изучил алгоритм поиска максимума и минимума, провел исследование и с помощью метода наименьших квадратов построил линейную зависимость и нашел коэффициент парной корреляции.