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 Выполние линейной регрессии и рассчет 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:

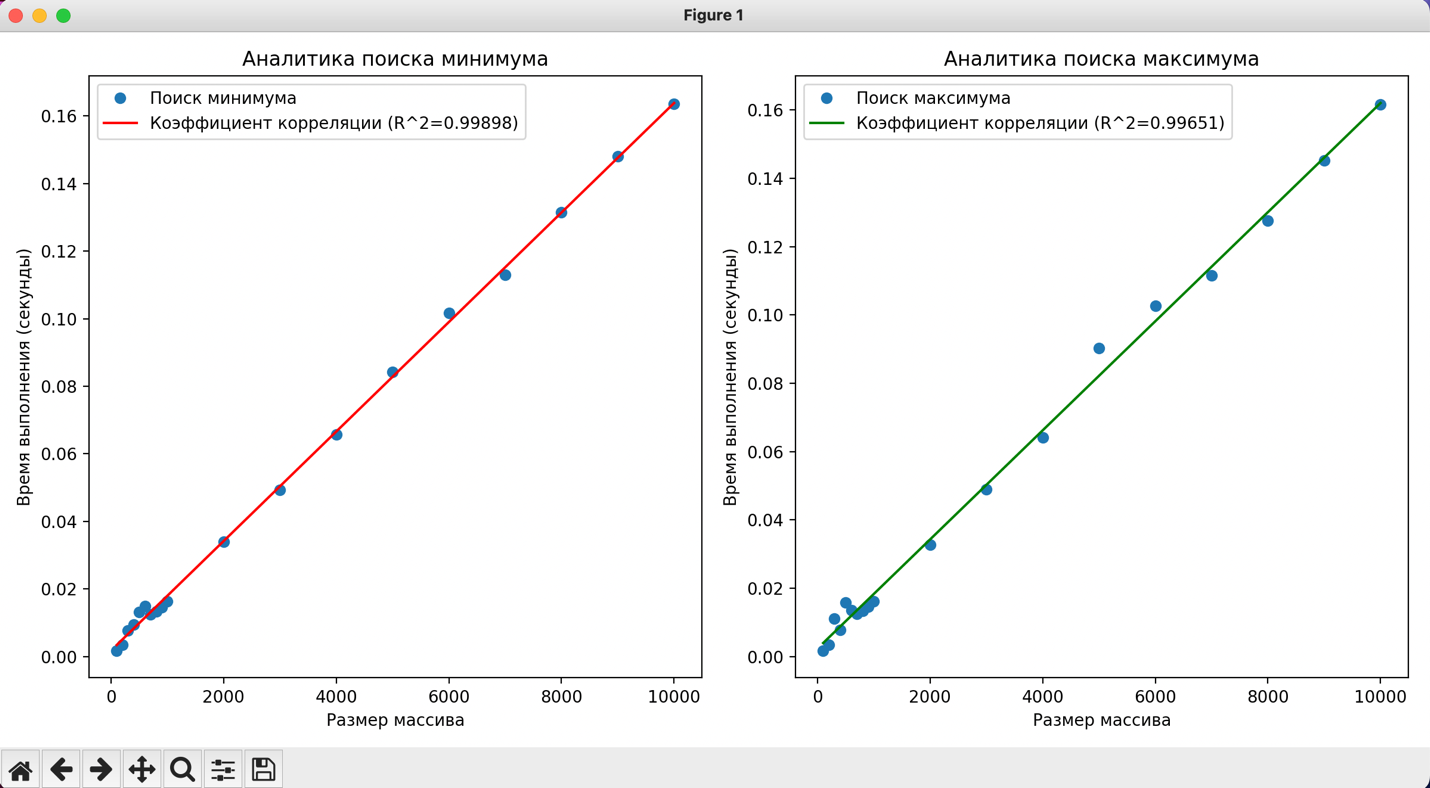


Рисунок 1 – Графики с аналитикой

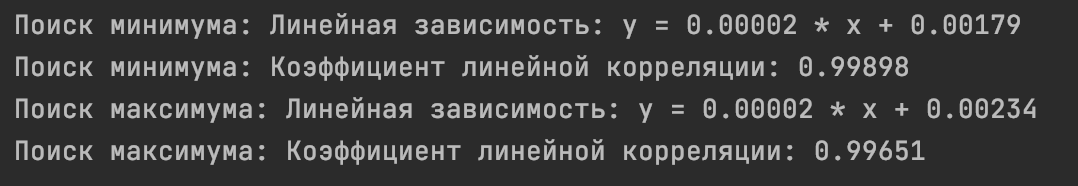


Рисунок 2 – Полученные данные

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