

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 №2
дисциплины «Алгоритмизация»

Выполнила:
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: Изучение алгоритма линейного поиска и анализ его работы в специфичных случаях.

Цель: изучить алгоритм линейного поиска, провести исследование в худшем и среднем случае и с помощью метода наименьших квадратов построить линейную зависимость и найти коэффициент парной корреляции.

Ход выполнения заданий

1) Написал программу для анализа алгоритма линейного поиска

Код программы:

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

# region Функции
def linear_search(arr, target): # функция для алгоритма линейного поиска
    for i in range(len(arr)):
        if arr[i] == target:
            return i
    return -1

def search_time_experiment(arr_size, target_position): # функция для
записывания времени экспериментов
    arr = list(range(arr_size))
    target = arr[target_position]

    execution_time = timeit.timeit(lambda: linear_search(arr, target),
number=1000)
    return execution_time

# endregion

# region Параметры для исследования
array_sizes = [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000,
                2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000] #
параметры для массивов
num_experiments = 5 # количество экспериментов
WorstCase_execution_times = [] # массив для времени выполнения худшего
случая
AvgCase_execution_times = [] # массив для времени выполнения среднего случая
# endregion

# Подготовка и проведение экспериментов
for size in array_sizes:
    WorstCase_times = []
    AvgCase_times = []

    for _ in range(num_experiments):
        WorstCase_time = search_time_experiment(size, size - 1)
        AvgCase_time = search_time_experiment(size, size // 2)

        WorstCase_times.append(WorstCase_time)
        AvgCase_times.append(AvgCase_time)
```

```

# region Вычисление среднего времени выполнения алгоритма
avg_worst_case_time = sum(WorstCase_times) / num_experiments
avg_average_case_time = sum(AvgCase_times) / num_experiments

WorstCase_execution_times.append(avg_worst_case_time)
AvgCase_execution_times.append(avg_average_case_time)
# endregion

# region Выполнение линейной регрессии и расчет коэффициента корреляции для
# обоих случаев.
x = np.array(array_sizes)
y_worst = np.array(WorstCase_execution_times)
y_avg = np.array(AvgCase_execution_times)

A = np.vstack([x, np.ones(len(x))]).T
a_worst, b_worst = np.linalg.lstsq(A, y_worst, rcond=None)[0]
a_avg, b_avg = np.linalg.lstsq(A, y_avg, rcond=None)[0]

correlation_coefficient_worst = np.corrcoef(x, y_worst)[0, 1] ** 2
correlation_coefficient_avg = np.corrcoef(x, y_avg)[0, 1] ** 2
# endregion

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

plt.subplot(1, 2, 1)
plt.plot(x, y_worst, 'o', label='Худший случай')
plt.plot(x, a_worst * x + b_worst, 'r', label=f'Линейная зависимость (R^2={correlation_coefficient_worst:.5f})')
plt.xlabel('Размер массива')
plt.ylabel('Время выполнения (секунды)')
plt.title('Аналитика худшего случая')
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(x, y_avg, 'o', label='Средний случай')
plt.plot(x, a_avg * x + b_avg, 'g', label=f'Линейная зависимость (R^2={correlation_coefficient_avg:.5f})')
plt.xlabel('Размер массива')
plt.ylabel('Время выполнения (секунды)')
plt.title('Аналитика среднего случая')
plt.legend()

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

# region Вывод в консоль
print(f"Худший случай: Линейная зависимость:  $y = \{a\_worst:.5f\} * x + \{b\_worst:.5f\}$ ")
print(f"Худший случай: Коэффициент корреляции:  $\{correlation\_coefficient\_worst:.5f\}$ ")

print(f"Средний случай: Средний случай:  $y = \{a\_avg:.5f\} * x + \{b\_avg:.5f\}$ ")
print(f"Средний случай: Коэффициент корреляции:  $\{correlation\_coefficient\_avg:.5f\}$ ")
# endregion

```

Результат выполнения:

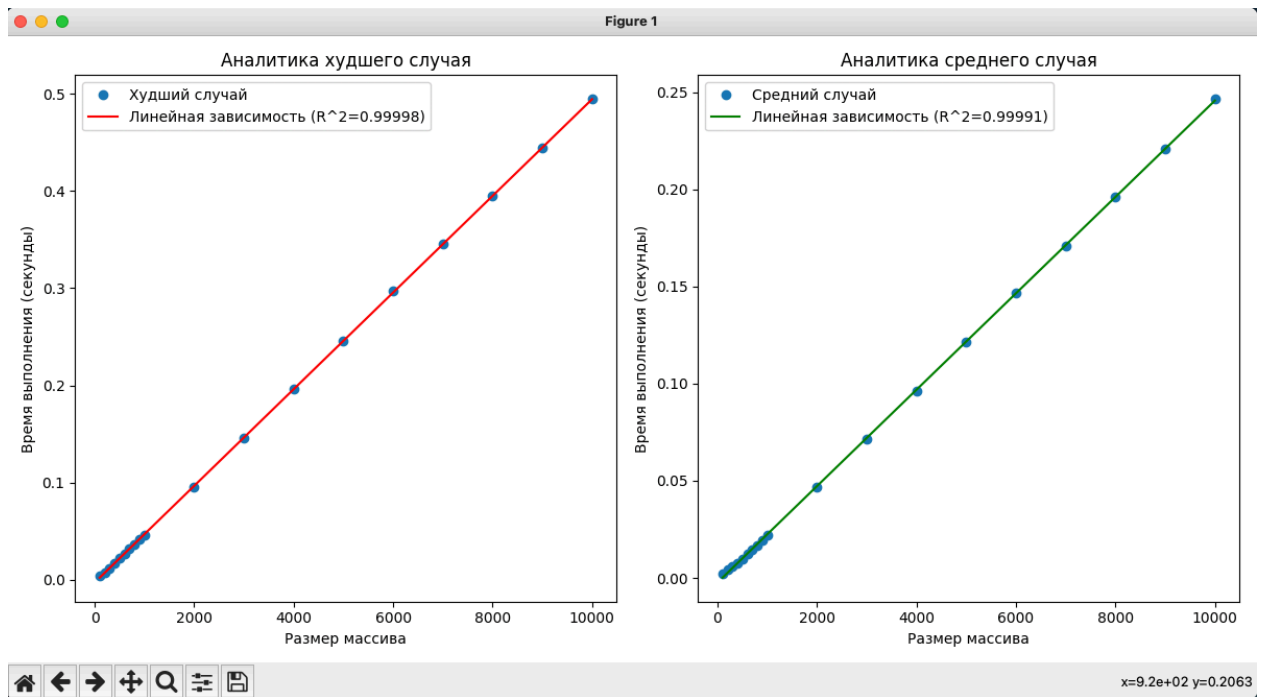


Рисунок 1 –Графики аналитики алгоритма линейного поиска

```
Худший случай: Линейная зависимость:  $y = 0.00005 * x + -0.00260$   
Худший случай: Коэффициент корреляции: 0.99987  
Средний случай: Средний случай:  $y = 0.00003 * x + -0.00233$   
Средний случай: Коэффициент корреляции: 0.99986
```

Рисунок 2 – Краткая сводка аналитики алгоритма линейного поиска

Вывод: изучил алгоритм линейного поиска, провел исследование в худшем и среднем случае и с помощью метода наименьших квадратов построил линейную зависимость и нашел коэффициент парной корреляции