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 №3
дисциплины «Алгоритмизация»

Выполнила:
Михеева Елена Александровна
2 курс, группа ИВТ-б-з-20-1,
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...,

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2023 г.

Порядок выполнения работы.

1. Изучен алгоритм линейного поиска. Была написана программа для нахождения элемента в массиве. Для расчетов времени работы алгоритма были исследованы худший и средний случаи. В худшем элемента не было в списке, а в среднем элемент находился в середине массива.

```python
import time


def linear_search(arr, value):
    for i in range(len(arr)):
        if arr[i] == value:
            return i
    return -1


array_lengths = list(range(100, 1000, 100))
value_worst = -1

for length in array_lengths:
    array_to_search = list(range(length))
    value_average = length // 2
    # Тестируем худший случай
    start_time_worst = time.time()
    linear_search(array_to_search, value_worst)
    end_time_worst = time.time()
    exe_time_worst = end_time_worst - start_time_worst

    # Тестируем средний случай
    start_time_average = time.time()
    linear_search(array_to_search, value_average)
    end_time_average = time.time()
    exe_time_average = end_time_average - start_time_average

    print(f"Время выполнения худшего случая: {exe_time_worst} сек.")
    print(f"Время выполнения среднего случая: {exe_time_average} сек.\n")

```

Рисунок 1. Программа для нахождения элемента в списке

2. Для исследования времени работы программы в худшем случае использовали метод наименьших квадратов.

											Сумма
t	100	200	300	400	500	600	700	800	900	1000	5500
y	0,000007868	0,000010967	0,000015020	0,000020266	0,000025034	0,000034850	0,000039829	0,000052697	0,000056996	0,000062141	0,000326
*t*t*	10000	40000	90000	160000	250000	360000	490000	640000	810000	1000000	3850000
*t*y*	0,000786781	0,00219345	0,00450611	0,008106232	0,012516975	0,02090991	0,027880638	0,042157837	0,051296059	0,062141388	0,232495
f(t)	0,000003452	0,000009922	0,000016392	0,000022862	0,000029332	0,000035802	0,000042272	0,000048742	0,000055212	0,000061682	

Рисунок 2. Таблица для метода наименьших квадратов для худшего случая

Полученная система уравнений для худшего случая:

$$\begin{cases} 3850000a + 5500b = 0{,}23249538; \\ 5500a + 10b = 0{,}00032567. \end{cases}$$

В ходе решения данной системы получили коэффициенты a = 0,000000065 и b = -0,000003018.

Получена функция y = 0,000000065t - 0,000003018.

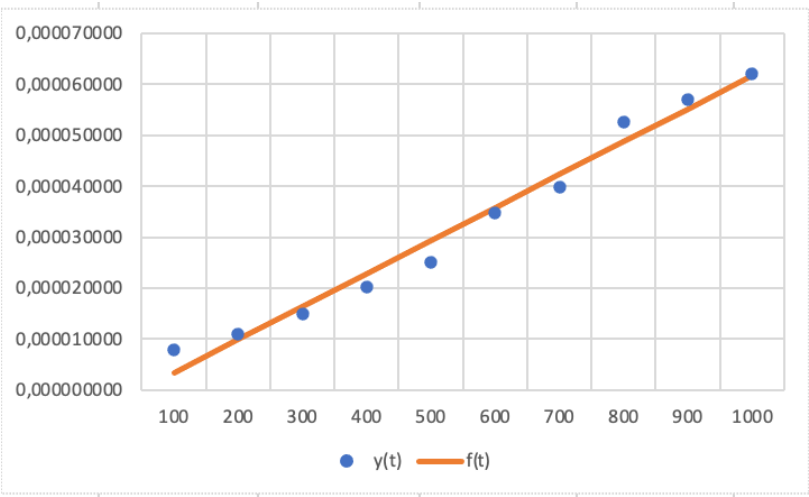

Рисунок 3. График y = 0,000000065t - 0,000003018

3. Для исследования времени работы программы в среднем случае использовали метод наименьших квадратов.

											Сумма
t	100	200	300	400	500	600	700	800	900	1000	5500
y	0,000000954	0,000004292	0,000010014	0,000006914	0,000010014	0,000011921	0,000014067	0,000017166	0,000019789	0,000021935	0,000117064
*t*t*	10000	40000	90000	160000	250000	360000	490000	640000	810000	1000000	3850000
*t*y*	9,54E-05	8,58E-04	3,00E-03	2,77E-03	5,01E-03	7,15E-03	9,85E-03	1,37E-02	1,78E-02	2,19E-02	0,08220673
f(t)	0,000001986	0,000004146	0,000006306	0,000008466	0,000010626	0,000012786	0,000014947	0,000017107	0,000019267	0,000021427	

Рисунок 4. Таблица для среднего случая

Полученная система уравнений для худшего случая:

$$\begin{cases} 3850000a + 5500b = 0{,}08220673; \\ 5500a + 10b = 0{,}000117064. \end{cases}$$

В ходе решения данной системы получили коэффициенты a = 0,000000022 и b = -0,000000175.

Получена функция y = 0,000000022t - 0,000000175

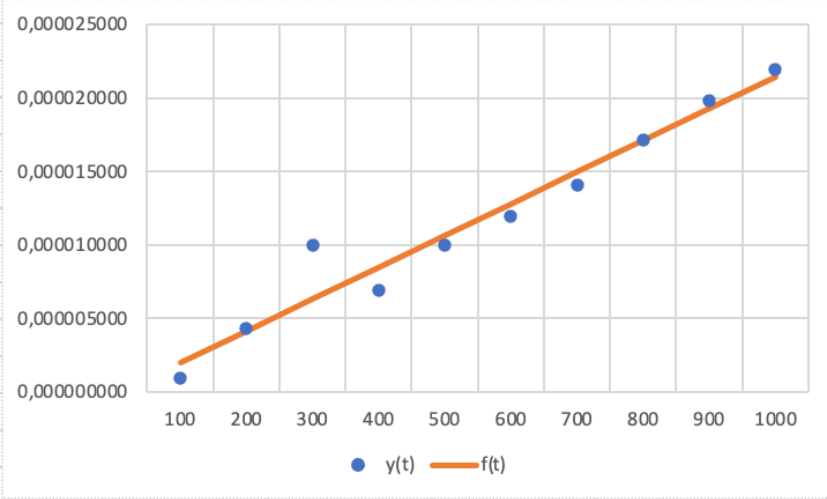

Рисунок 5. График y = 0,000000022t - 0,000000175

4. Был найден коэффициент корреляции для выявления связи между полученными методом измерения данными и массивом переменных, полученных из функции y = 0,000000022t - 0,000000175.

Коэффициент корелляции:	0,97538883

Рисунок 6. Коэффициент корреляции

Вывод: время выполнения алгоритма растет линейно в зависимости от количества элементов массива. Значение коэффициента корреляции близкое к 1 указывает на сильную линейную связь.