

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 №9**  
**дисциплины «Алгоритмизация»**

Выполнил:  
Гайчук Дарья Дмитриевна  
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. Написала программу сравнения функций поиска: линейный и бинарный поиск. Сравнила их со встроенной в Python функцией бинарного поиска `bisect.bisect_left()`.

```
2 usage new *
def linear_search(a, b):
    len_mass = len(a)
    for i in range(len_mass):
        if b == a[i]:
            return i
    return -1

1 usage new *
def binary_search(a, k):
    l, r = 0, len(a)
    while l < r:
        m = (l+r) // 2
        if a[m] == k:
            return m
        elif a[m] > k:
            r = m
        else:
            l = m + 1
    return -1

1 usage new *
def bisect_search(arr, target):
    index = bisect.bisect_left(arr, target)
    if index != len(arr) and arr[index] == target:
        return index
    return -1
```

Рисунок 1. Реализация трех алгоритмов

```
58
1 usage new *
59 def plot_graph(b, c, namegraph, bool_l):
60     plt.scatter(b, c, s=5)
61     if bool_l:
62         aur, bur = find_coeffs_line(b, c)
63         y_line = aur * np.array(b) + bur
64     else:
65         aur, bur = find_coeffs_bin(b, c)
66         y_line = log_n(np.array(b), aur, bur)
67     plt.plot(*args: b, y_line, color='red')
68     plt.title(namegraph + " случай")
69     plt.xlabel("Размер массива")
70     plt.ylabel("Время работы функции")
71
72
```

Рисунок 2. Создание графика

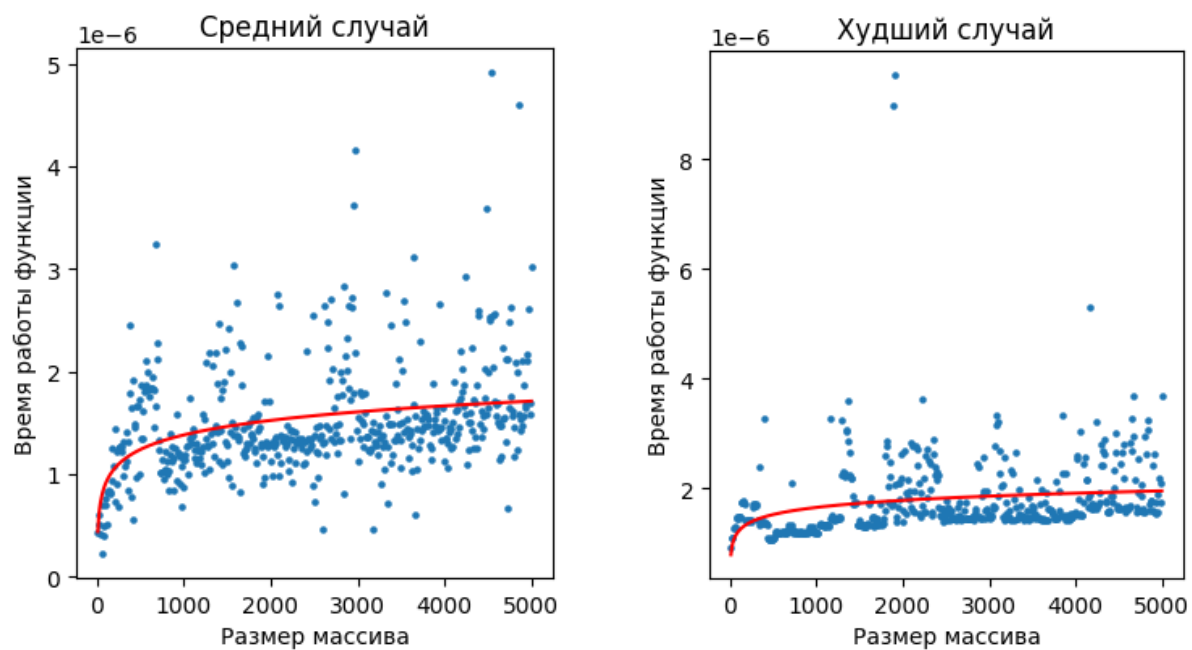


Рисунок 3. Бинарный поиск

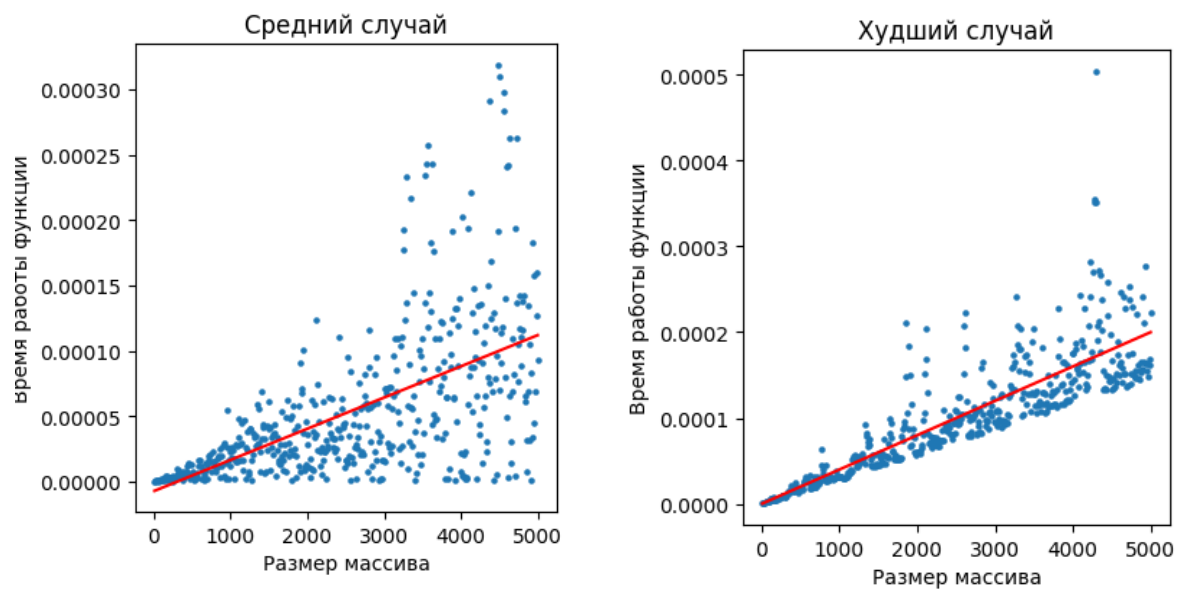


Рисунок 4. Линейный поиск

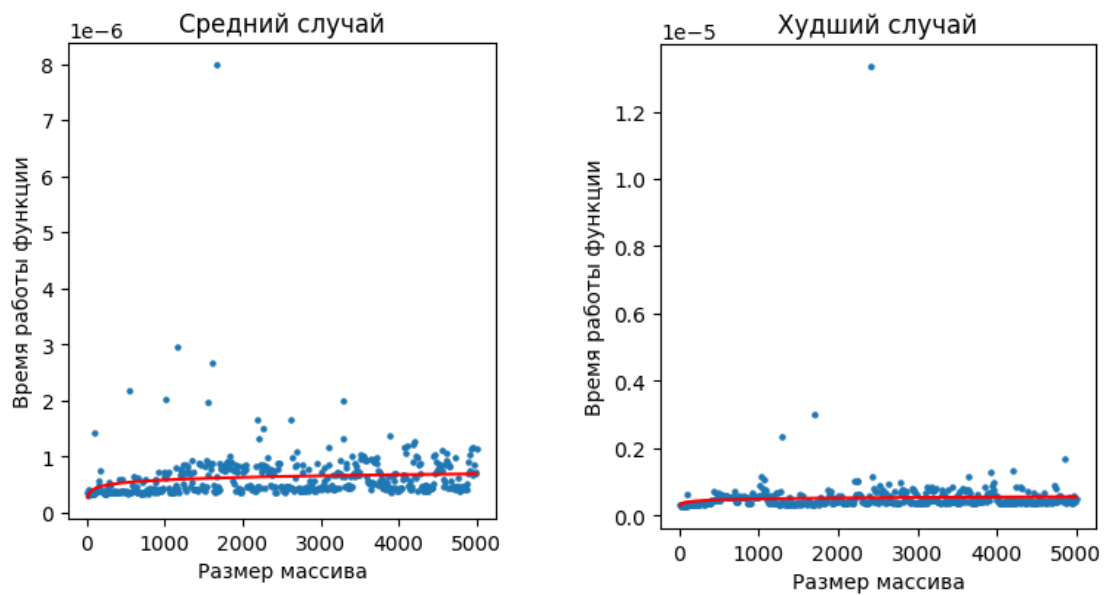


Рисунок 5. Встроенная функция Python

Вывод: в ходе выполнения лабораторной работы были исследованы функции поиска в двух случаях: среднем и худшем. Лучшей оказалась функция, встроенная в Python: `bisect.bisect_left()`, а худшей – линейный поиск. Для эффективного поиска элемента в массиве лучше всего воспользоваться встроенной функцией в Python `bisect.bisect_left()`.