

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:
Кожуховский Виктор Андреевич
1 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 Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Порядок выполнения работы:

Написал программу поиска наименьшего и наибольшего элемента в массиве, автоматического заполнения массива, расчёта тысячи точек, показывающих время поиска элемента в массиве в худшем и среднем случае, вывода графиков, составленных из этих точек, и подсчета корреляции:

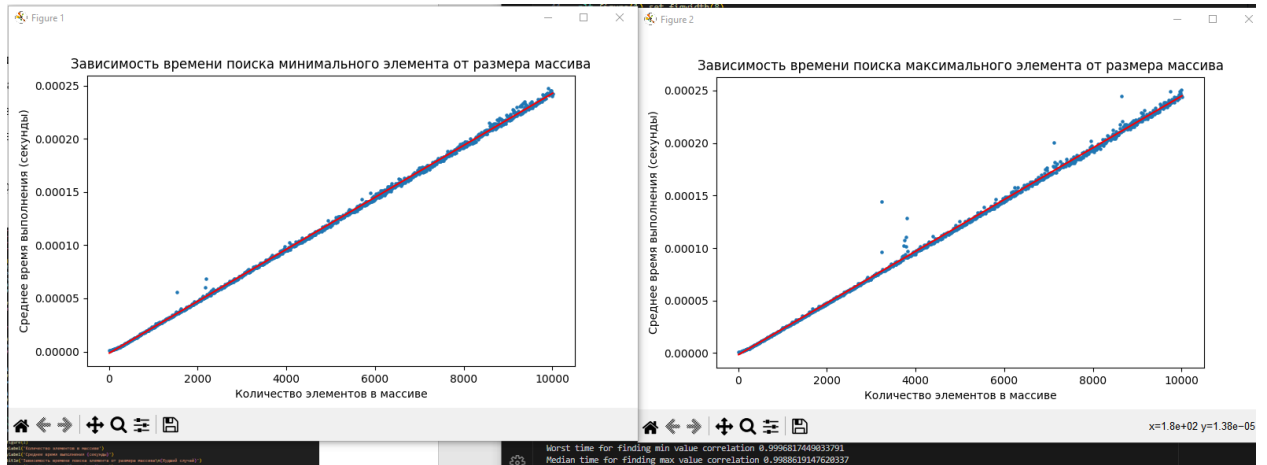


Рисунок 1. Графики времени поиска наименьшего и наибольшего элемента в массиве в худшем и среднем случае и вывод корреляции

```
1 import timeit
2 import random
3 import matplotlib.pyplot as plt
4 import numpy as np
5
6 a = {}
7 worstTimeFindMin = {}
8 worstTimeFindMax = {}
9 GraphStuff = [i for i in range(10, 10020, 10)]
10 StuffForLisWorstFindMin = {}
11 StuffForLisWorstFindMax = {}
12
13 def findMin():
14     min = 999999
15     for i in range(len(a)):
16         if a[i] < min:
17             min = a[i]
18     return min
19
20 def findMax():
21     max = -999999
22     for i in range(len(a)):
23         if a[i] > max:
24             max = a[i]
25     return max
26
27 def fillArr(numOfEl):
28     a.clear()
29     for i in range(numOfEl):
30         a[i] = random.randint(0, 100000)
31
32 for i in range(10, 10020, 10):
33     fillArr(i)
34     worstTimeFindMin[i] = (timeit.timeit(lambda: findMin(), number = 10)) / 10
35
36 for i in range(10, 10020, 10):
37     fillArr(i)
38     worstTimeFindMax[i] = (timeit.timeit(lambda: findMax(), number = 10)) / 10
39
40 A = np.vstack([GraphStuff, np.ones(len(GraphStuff))]).T
41 y = np.array(list(worstTimeFindMin.values()))[:, np.newaxis]
42 alpha = np.dot((np.dot(np.linalg.inv(np.dot(A.T, A)), A.T)), np.array(list(worstTimeFindMin.values())))) # Базис из книги "Python Programming And Numerical Methods: A S
43
44 plt.figure(1).set_figwidth(8)
45 plt.xlabel('Количество элементов в массиве')
46 plt.ylabel('Среднее время выполнения (секунды)')
47 plt.title('Зависимость времени поиска минимального элемента от размера массива')
48 plt.scatter(GraphStuff, worstTimeFindMin.values(), s=5)
49 plt.grid(False)
50 plt.plot(GraphStuff, alpha[0]*np.array(list(GraphStuff)) + alpha[1], 'r')
51
52 A = np.vstack([GraphStuff, np.ones(len(GraphStuff))]).T
53 y = np.array(list(worstTimeFindMax.values()))[:, np.newaxis]
54 alpha = np.dot((np.dot(np.linalg.inv(np.dot(A.T, A)), A.T)), np.array(list(worstTimeFindMax.values()))))
55
56 plt.figure(2).set_figwidth(8)
57 plt.xlabel('Количество элементов в массиве')
58 plt.ylabel('Среднее время выполнения (секунды)')
59 plt.title('Зависимость времени поиска максимального элемента от размера массива')
60 plt.scatter(GraphStuff, worstTimeFindMax.values(), s=5)
61 plt.grid(False)
62 plt.plot(GraphStuff, alpha[0]*np.array(list(GraphStuff)) + alpha[1], 'r')
63
64 print('Worst time for finding min value correlation', np.corrcoef(GraphStuff, list(worstTimeFindMin.values()))[0, 1])
65 print('Median time for finding max value correlation', np.corrcoef(GraphStuff, list(worstTimeFindMax.values()))[0, 1])
66
67 plt.show()
```

Рисунок 2. Код программы

Вывод: в результате выполнения лабораторной работы был изучен алгоритм линейного поиска наименьшего и наибольшего элемента в массиве и проведено исследование зависимости времени этого поиска от количества элементов в массиве.