

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 №5
дисциплины «Алгоритмизация»

Выполнил:
Кенесбаев Хилол Куат улы
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. Написал программу, которая выводит в виде графиков зависимость времени выполнения функций пузырьковой сортировки от размера массива в двух случаях: среднем и худшем. Также построил график среднего случая из средних значений времени выполнения функции для каждого x , отметил среднее отклонение и по методу средних квадратов построил параболу, в худшем случае также построил параболу по методу меньших квадратов:

```
main.py x
C:\Users\HP-PC\Desktop\study\Алгоритмизация 5\алгоритмизация 5\prog> main.py
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import itertools
5 import math
6 import random as rnd
7 import matplotlib.pyplot as plt
8 import numpy as np
9 import timeit
10
11
12 def bubble_sort(arr):
13     n = len(arr)
14     for i in range(n-1):
15         for j in range(n-1-i):
16             if arr[j] > arr[j+1]:
17                 arr[j], arr[j+1] = arr[j+1], arr[j]
18     return arr
19
20
21 def coeffs(xs, ys):
22     n = len(xs)
23     s1 = sum(xs)
24     s2 = sum(x**2 for x in xs)
25     s3 = sum(x**3 for x in xs)
26     s4 = sum(x**4 for x in xs)
27     sy0 = sum(ys)
28     sy1 = sum(xs[i]*ys[i] for i in range(n))
29     sy2 = sum(xs[i]**2*ys[i] for i in range(n))
30     matrixx = [[n, s1, s2], [s1, s2, s3], [s2, s3, s4]]
31     matrixy = [[sy0], [sy1], [sy2]]
32     x = np.linalg.solve(matrixx, matrixy)
33     return x[2][0], x[1][0], x[0][0]
34
35
36 def create_graph(x, y, namegraph):
37     plt.scatter(x, y, s=5)
38     plt.title(namegraph)
39     plt.xlabel('Размер массива')
40     plt.ylabel('Время работы функции')
41
42
43 if __name__ == '__main__':
44     count = 50
45     x1 = [i for i in range(10, count*10+1, 10)]
46     x2 = [i*30 for i in x1]
47     xgraph = list(itertools.chain.from_iterable(x2))
48     randmax = 1000000
49     timesred = []
50     timeshud = []
51
52     for i in xgraph:
53         sred = [rnd.randint(0, randmax) for j in range(i)]
54         timesred.append(timeit.timeit(lambda: bubble_sort(sred), number=1))
55
56         hud = [j for j in range(1, 0, -1)]
57         timeshud.append(timeit.timeit(lambda: bubble_sort(hud), number=1))
58
59     timesred = [timesred[i: i+30] for i in range(0, len(timesred), 30)]
60     e = [(1/30*(sum(timesred[i])))) for i in range(len(timesred))]
61
62     signavalue = [sum([(timesred[k][j] - e[k])**2 for j in range(30)]) for k in range(count)]
63     sigma = [math.sqrt(1/20*signavalue[k]) for k in range(count)]
64     a, b, c = coeffs(x1, e)
65     yred = a * np.array(x1) ** 2 + b * np.array(x1) + c
66
67     a, b, c = coeffs(x1, timeshud)
68     yhud = a * np.array(x1) ** 2 + b * np.array(x1) + c
69
70     name = 'Средний случай'
71     create_graph(xgraph, timesred, name)
72
73     name = 'Средний случай'
74     plt.figure(name)
75     plt.errorbar(x1, e, yerr=sigma, fmt='none', capsize=2)
76     plt.plot(x1, yred, color='green', linewidth=2)
77     create_graph(x1, e, name)
78
79     name = 'Худший случай'
80     plt.figure(name)
81     plt.plot(x1, yhud, color='green', linewidth=2)
82     create_graph(x1, timeshud, name)
83
84     plt.show()
```

Рисунок 1 Код программы

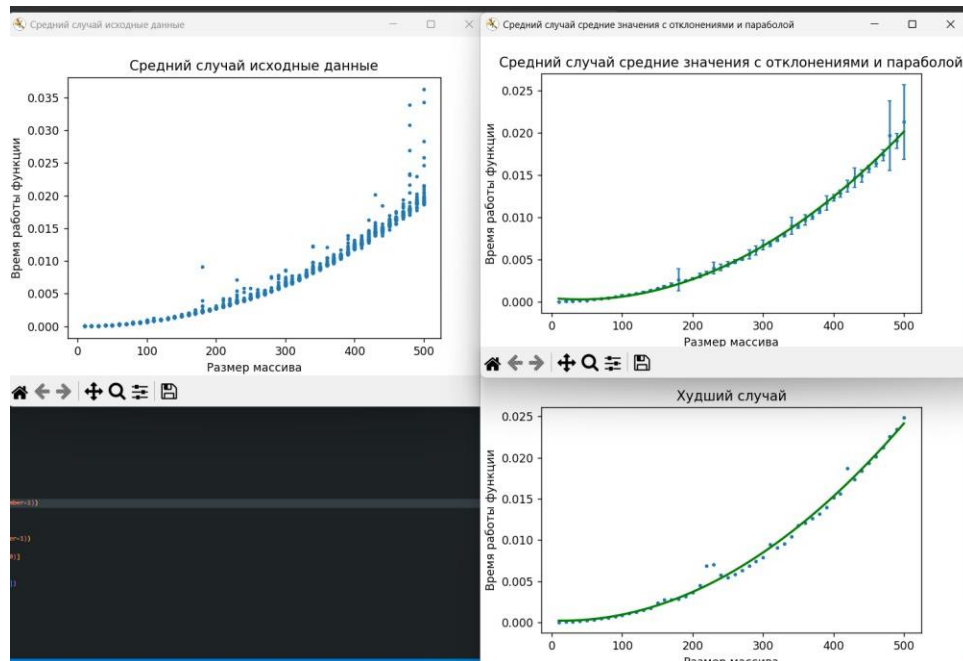


Рисунок 2 График программы

Вывод: В ходе выполнения лабораторной работы был проведен анализ зависимости времени выполнения функции пузырьковой сортировки от размера массива в худшем и среднем случаях. Из полученных результатов можно сделать следующий вывод: время работы функции пузырьковой сортировки экспоненциально растет с увеличением размера массива.