

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 ПРАКТИЧЕСКОЙ РАБОТЕ №1.5
дисциплины «Алгоритмизация»

Выполнил:
Степанов Леонид Викторович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение
средств вычислительной
техники и автоматизирование
систем», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., канд. техн. наук,
доцент, доцент кафедры
инфокоммуникаций

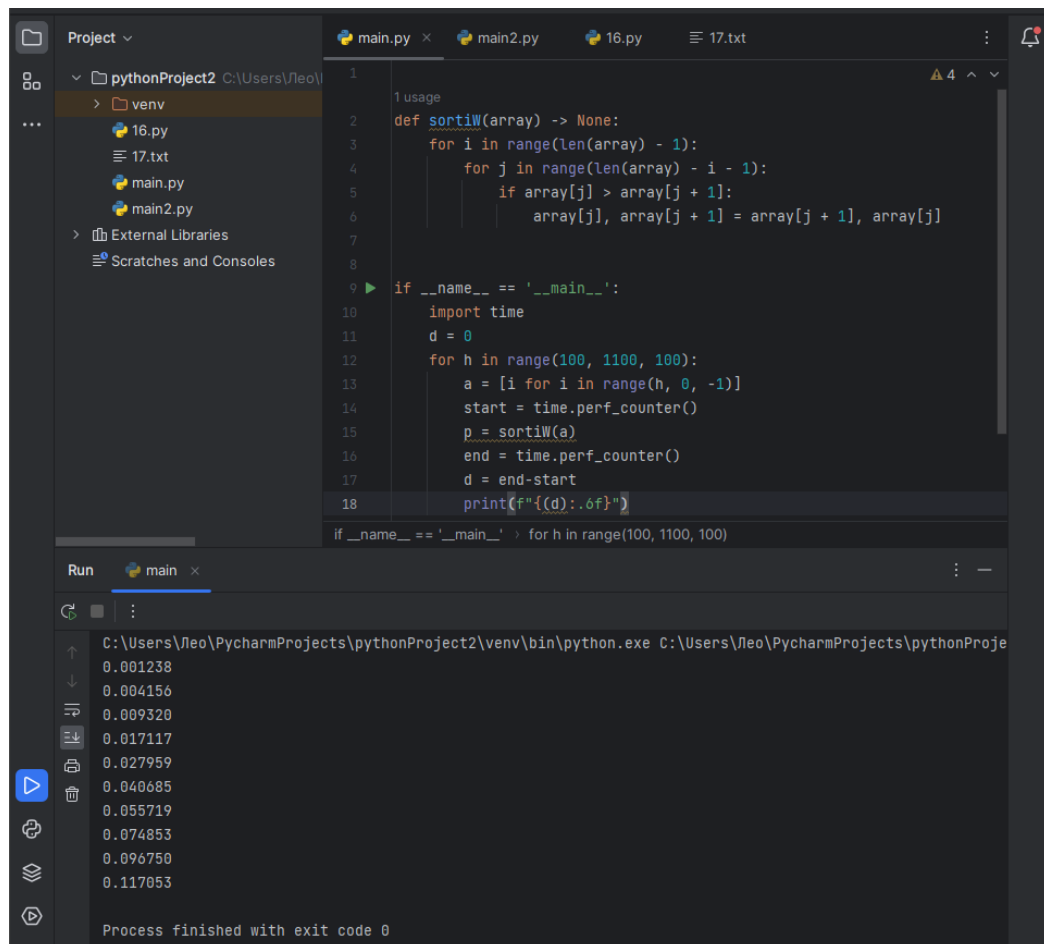
(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Порядок выполнения работы:

1. Написал программу (puzW.py) в которой измеряется время выполнения пузырьковой сортировки, метод пузырька заключается в том, что любые два подряд идущие элементы сравниваются и меньшее ставит левее большего и так с каждым элементом массива. Худший случай заключается в том, что элементы расположены в убывании.



```
1 usage
2 def sortiW(array) -> None:
3     for i in range(len(array) - 1):
4         for j in range(len(array) - i - 1):
5             if array[j] > array[j + 1]:
6                 array[j], array[j + 1] = array[j + 1], array[j]
7
8
9 if __name__ == '__main__':
10     import time
11     d = 0
12     for h in range(100, 1100, 100):
13         a = [i for i in range(h, 0, -1)]
14         start = time.perf_counter()
15         p = sortiW(a)
16         end = time.perf_counter()
17         d = end-start
18         print(f"{(d):.6f}")
19
20 if __name__ == '__main__':
21     for h in range(100, 1100, 100):
```

Run console output:

```
0.001238
0.004156
0.009320
0.017117
0.027959
0.040685
0.055719
0.074853
0.096750
0.117053
Process finished with exit code 0
```

Рисунок 1 – Результат выполнения puzW.py

n	100	200	300	400	500	600	700	800	900	1000	5500
t	0,001032	0,00411	0,009329	0,01697	0,027323	0,040453	0,0572	0,07999	0,094113	0,121992	0,452512
n^2	10000	40000	90000	160000	250000	360000	490000	640000	810000	1000000	3850000
n*t	0,1032	0,822	2,7987	6,788	13,6615	24,2718	40,04	63,992	84,7017	121,992	359,1709

Рисунок 2 – Таблица данных puzW.py в Exel

При помощи метода наименьших квадратов вывели систему уравнений:
 $3850000a + 5500b = 359,1709$ и $5500a + 10b = 0,4525$, решив которую мы нашли график функции: $y = 0,0001337x - 0,028275$



Рисунок 3 – График функции $y = 0,0001337x - 0,028275$

2. Написал программу (puzSR.py), в которой измеряется время использования метода пузырьковой сортировки в среднем случае, когда элементы массива вводятся рандомно.

```

1  if __name__ == '__main__':
2      from random import randint
3      from math import sqrt
4      import time
5      e = []
6      sigma = []
7      d = 0
8      q = 0
9      for i in range(100, 1000, 100):
10         for l in range(1, 31):
11             a = [0 * t for t in range(i)]
12             for k in range(i):
13                 a.append(randint(100, 1000))
14             start = time.perf_counter()
15             p = sortiSR(a)
16             end = time.perf_counter()
17             d += (end-start)/30
18             q += ((end - start) - (d*30)) ** 2
19         e.append(d)
20         sigma.append(sqrt(1/30*q))
21         d = 0
22         q = 0
23     print(e)
24     print(sigma)

```

Run main

```

C:\Users\Leo\PycharmProjects\pythonProject2\venv\bin\python.exe C:\Users\Leo\PycharmProjects\pythonProje
[0.0019386466665916185, 0.008649859999996861, 0.020534040000044728, 0.036258316666620286, 0.04971703000001
[0.032675500779330256, 0.1477909157325066, 0.3416871082766057, 0.6196776964030715, 0.8506845969895279, 1
Process finished with exit code 0

```

Рисунок 4 – Результат выполнения программы puzSR.py

	A	B	C	D	E	F	G	H	I	J	K
1	n	100	200	300	400	500	600	700	800	900	4500
2	e[n]	0,001714372	0,007520686	0,017635707	0,032387486	0,050600438	0,075399703	0,102152693	0,136590624	0,173666507	0,597668217
3	n^2	10000	40000	90000	160000	250000	360000	490000	640000	810000	2850000
4	n^3	1000000	8000000	27000000	64000000	125000000	216000000	343000000	512000000	729000000	2025000000
5	n^4	100000000	1600000000	8100000000	25600000000	62500000000	1,296E+11	2,401E+11	4,096E+11	6,561E+11	1533300000000,00
6	n^2e	17,14372414	300,8274483	1587,213621	5181,997793	12650,10948	27143,89324	50054,81962	87417,99945	140669,8706	325023,8751709
7	ne[n]	0,171437241	1,504137241	5,290712069	12,95499448	25,30021897	45,23982207	71,50688517	109,2724993	156,2998562	427,5405628
8	N	-0,019402	0,0020555	0,023513	0,0449705	0,066428	0,0878855	0,109343	0,1308005	0,152258	
9	sigma	0,0326755	0,1477909	0,3416871	0,619677	0,867854	1,2736104	1,7704698	2,1959463	2,85037173	
10	3sigma	0,0980265	0,4433727	1,0250613	1,859031	2,603562	3,8208312	5,3114094	6,5878389	8,55111519	

Рисунок 5 – Таблица данных puzSR.py

При помощи метода наименьших квадратов вывели систему уравнений: $1533399900000000a + 2025000000b + 285000c = 325023,8751709$, $2025000000a + 2850000b + 4500c = 427,5405628$, $2850000a + 4500b + 9c = 0,597668217$, решив которую мы нашли график функции: $y = (-6,38E - 11)^2 \cdot x + 0,00021b - 0,0408595$

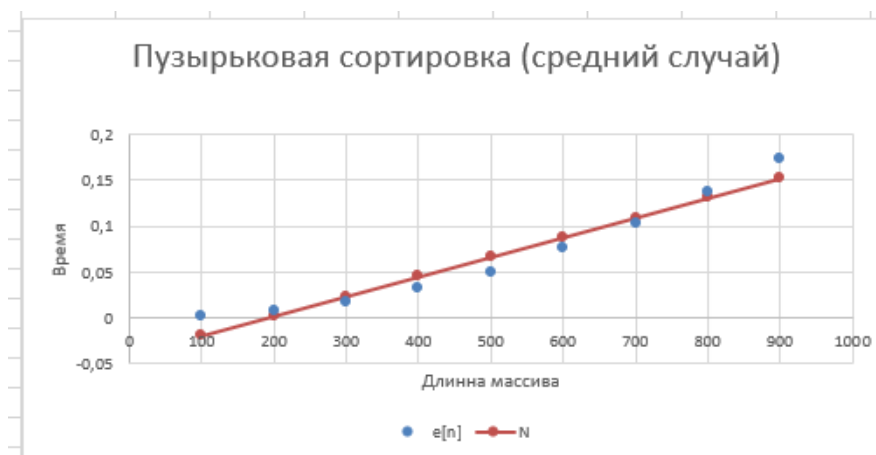


Рисунок 6 – График функции $y = (-6,38E - 11)^2 \cdot x + 0,00021b - 0,0408595$

Далее наложили на график пределы погрешностей:

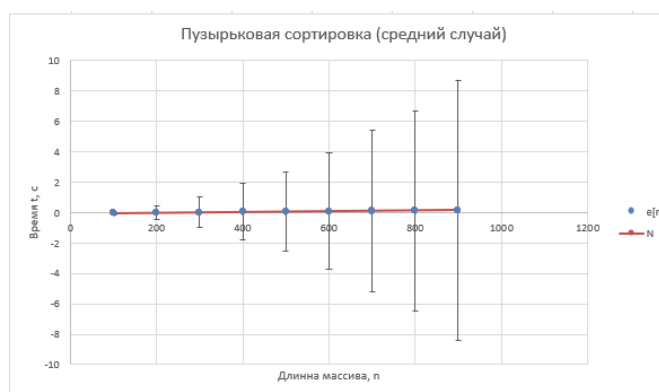


Рисунок 7 – График функции $y = (-6,38E - 11)^2 \cdot x + 0,00021b - 0,0408595$

Вывод: в результате проделанной работы было выяснено, что количество элементов в массиве влияет на время сортировки и O-большое пузырьковой сортировки – $O(n^2)$.