

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:
Михеева Елена Александровна
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. канд. техн. наук,
доцент, доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

Порядок выполнения работы:

1. Была написана программа, в которой измеряется время выполнения метода пузырьковой сортировки для худшего и среднего случаев. Были построены графики на основе метода наименьших квадратов для обоих случаев.

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def bubble_sorting(arr):
6      for i in range((len(arr)) - 1):
7          for j in range(len(arr)-i-1):
8              if arr[j] > arr[j+1]:
9                  arr[j], arr[j+1] = arr[j+1], arr[j]
10
11
12  if __name__ == '__main__':
13      array_lengths = list(range(100, 1100, 100))
14
15      for length in array_lengths:
16          import random
17          import time
18
19          array_to_sort = list(range(length))
20
21          # Худший случай
22          array_to_sort.reverse()
23          start_time_worst = time.time()
24          bubble_sorting(array_to_sort)
25          end_time_worst = time.time()
26          exe_time_worst = end_time_worst - start_time_worst
27
28          # Средний случай
29          random.shuffle(array_to_sort)
30          start_time_average = time.time()
31          bubble_sorting(array_to_sort)
32          end_time_average = time.time()
33          exe_time_average = end_time_average - start_time_average
34
35          print(f"Время выполнения худшего случая: {exe_time_worst} сек.")
36          print(f"Время выполнения среднего случая: {exe_time_average} сек.\n")
37
```

Рисунок 1. Программа bubble_sort.py

	Худший случай										Сумма
n	100	200	300	400	500	600	700	800	900	1000	5500
$n*n$	10000	40000	90000	160000	250000	360000	490000	640000	810000	1000000	3850000
n^3	1000000	8000000	27000000	64000000	125000000	216000000	343000000	512000000	729000000	1000000000	3025000000
n^4	100000000	1600000000	8100000000	25600000000	62500000000	1,296E+11	2,401E+11	4,096E+11	6,561E+11	1E+12	2,5333E+12
t	0,000762939	0,002502918	0,005816698	0,010445833	0,016956091	0,024630070	0,035137177	0,047701120	0,061561108	0,073282957	0,278796911
$n*t$	7,63E-02	5,01E-01	1,75E+00	4,18E+00	8,48E+00	1,48E+01	2,46E+01	3,82E+01	5,54E+01	7,33E+01	221,2011814
$t*n^2$	7,629394531	100,1167297	523,5028267	1671,333313	4239,022732	8866,8251	17217,21649	30528,717	49864,49718	73282,9571	186301,8178940

Рисунок 2. Таблица для худшего случая

2. Для построения графика для худшего случая составили систему уравнений:

$$\begin{cases} 10a + 5500b + 3850000c = 0,27879691 \\ 5500a + 3850000b + 3025000000c = 221,2011814 \\ 3850000a + 3025000000b + 2533300000000c = 186301,8179 \end{cases}$$

Матрица:

10	5500	3850000
5500	3850000	3025000000
3850000	3025000000	2,5333E+12

Обратная:

1,383333333	-0,00525	4,16667E-06
-0,00525	2,41288E-05	-2,08333E-08
4,16667E-06	-2,0833E-08	1,89394E-11

x

St

0,278796911
221,2011814
186301,8179

=

a

0,000620433
-7,65527E-06
8,17394E-08

a

b

c

Уравнение для графика:

$$y = 0,000620433 - 0,0000076553 \cdot n + 0,0000000817394 \cdot n^2$$

Рисунок 3. Получение уравнения

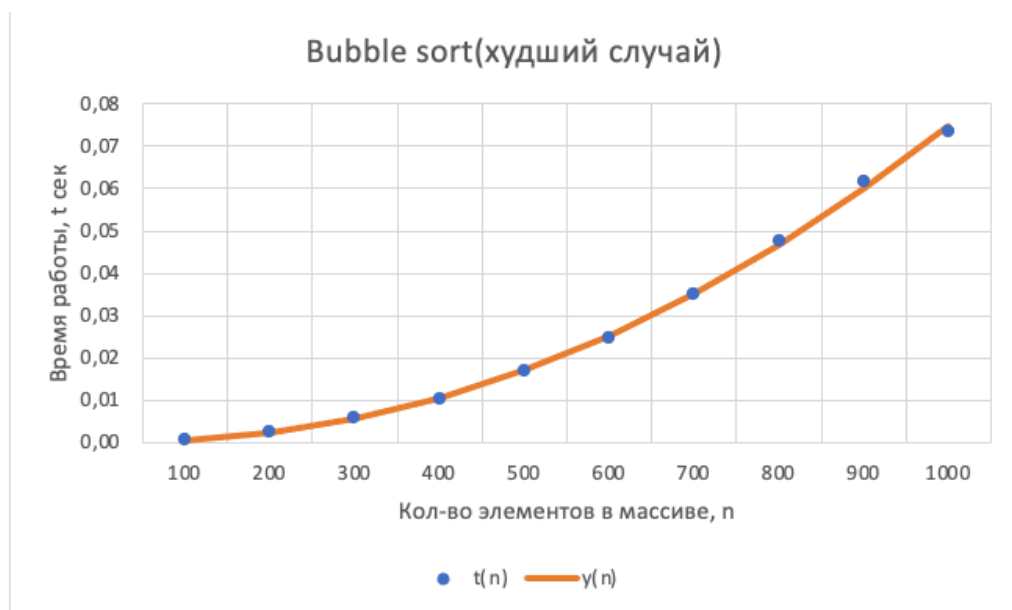


Рисунок 4. График для худшего случая

3. Для среднего случая составили таблицу данных и на ее основе построили график.

	Средний случай									
<i>n</i>	100	200	300	400	500	600	700	800	900	1000
	0,000569820	0,001962900	0,004340887	0,009188890	0,014261723	0,020461798	0,027453899	0,039705992	0,048637857	0,064547043
	0,000512123	0,001946211	0,004658937	0,008105993	0,013349771	0,019731283	0,027201176	0,039402008	0,045320988	0,056460619
	0,000514746	0,0019629	0,004372835	0,008116961	0,01356411	0,019729137	0,026742935	0,036796093	0,046324015	0,057109833
	0,000483036	0,001912117	0,004653931	0,008008003	0,013092995	0,019024134	0,026655912	0,035430908	0,044584990	0,057020187
	0,000503063	0,001895905	0,004563332	0,008319855	0,012847185	0,019180059	0,027467012	0,036319971	0,045594215	0,056579828
	0,000477791	0,002058983	0,004410028	0,007839918	0,013560057	0,019517899	0,026762009	0,036130905	0,045235872	0,05749011
	0,000510931	0,002053261	0,004527092	0,008152008	0,013210058	0,019516945	0,027120113	0,036030769	0,044948816	0,05724597
	0,000515938	0,001935005	0,004886866	0,008337736	0,013067007	0,019242048	0,027266026	0,036077976	0,045747042	0,055758238
	0,000495195	0,002852917	0,004258871	0,008172989	0,013023138	0,019572973	0,027047873	0,036984205	0,04536128	0,057779789
	0,00048995	0,001919031	0,004275322	0,008196831	0,01286602	0,019109964	0,027068853	0,03592205	0,04572916	0,056704998
	0,000558138	0,00192976	0,004634142	0,008352995	0,013389111	0,019041061	0,02801609	0,036806822	0,045560122	0,056684017
	0,000486851	0,001940966	0,004431009	0,00790906	0,013519287	0,02004385	0,026983023	0,036000967	0,045923233	0,057034254
	0,000488043	0,001924038	0,004433155	0,007756948	0,013364792	0,020117044	0,027466059	0,036831141	0,046939135	0,060996056
	0,000519991	0,001912117	0,004601717	0,009648903	0,013232937	0,020020962	0,026433229	0,035911083	0,045897245	0,057013988
	0,000506163	0,001927853	0,004477978	0,008643866	0,013572216	0,019851923	0,029003143	0,036561966	0,04527998	0,056951046
	0,000826836	0,002158165	0,004364967	0,008006811	0,013216972	0,020364046	0,027354956	0,036901951	0,046002865	0,056559801
	0,000781059	0,001999855	0,004603148	0,008048773	0,012825012	0,019585133	0,028385878	0,036036015	0,045670986	0,056921005
	0,000475883	0,001933098	0,004703999	0,008426905	0,016289949	0,019747019	0,027265787	0,037075996	0,045013905	0,056719065
	0,000509024	0,00193119	0,004462957	0,008167028	0,013420105	0,019616842	0,026734114	0,036379099	0,045496941	0,056633949
	0,000488997	0,002410889	0,004493952	0,007754803	0,013744831	0,020866871	0,027137041	0,035807133	0,045927286	0,057397842
	0,000519991	0,002169132	0,005013943	0,008618116	0,014065027	0,020089865	0,027849913	0,036665201	0,046290874	0,060863018
	0,000510216	0,002006054	0,004535198	0,008105993	0,013745785	0,019676924	0,027634859	0,037244797	0,052295923	0,056860924
	0,000497103	0,001893044	0,004732847	0,008075953	0,013538837	0,01961422	0,026964426	0,036469936	0,045938253	0,057599068
	0,000522614	0,001905203	0,004683018	0,008061171	0,012875795	0,019396067	0,027024984	0,036468267	0,045560837	0,056896925
	0,000586033	0,002130747	0,004230022	0,008301973	0,013060093	0,019245863	0,027425766	0,036362171	0,044986963	0,056914806
	0,000488997	0,001276016	0,005029917	0,007980108	0,01334095	0,019156218	0,026737928	0,035635948	0,050348043	0,05748105
	0,000491142	0,00197196	0,005628109	0,008394003	0,013156891	0,019273281	0,027583838	0,037125826	0,045196772	0,056746244
	0,000507832	0,001068807	0,005670309	0,013781071	0,013678074	0,020373821	0,027751923	0,035088062	0,045443058	0,056937933
	0,000498056	0,002093792	0,004337788	0,007960796	0,013731956	0,020308018	0,027235031	0,036026955	0,045471907	0,056503773
	0,000504017	0,001904011	0,004432917	0,008290052	0,014237165	0,019601822	0,028192997	0,036316156	0,045477867	0,060141087

Рисунок 5. Таблица данных для среднего случая

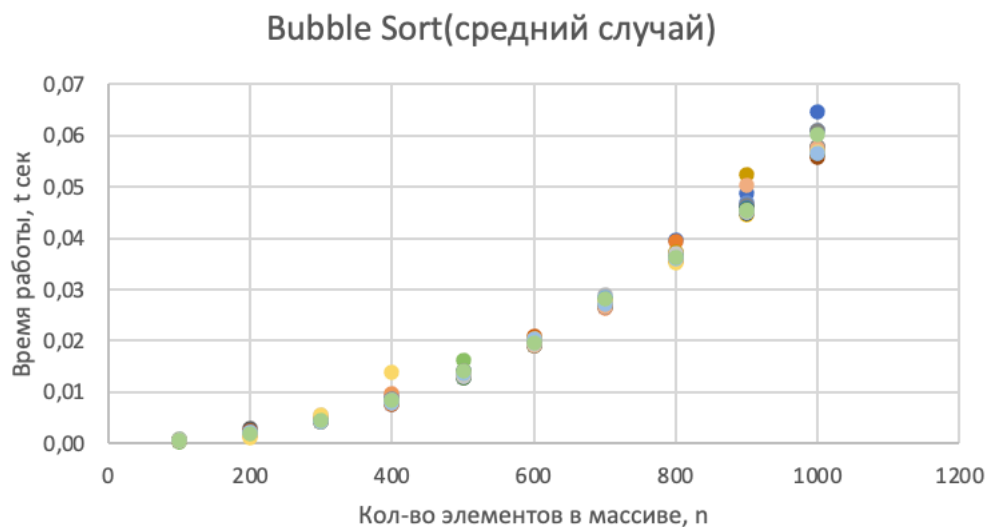


Рисунок 6. Диаграмма по полученным данным

С помощью метода наименьших квадратов составили систему уравнений для среднего случая:

$$\begin{cases} 10a + 5500b + 3850000c = 0,216238874 \\ 5500a + 3850000b + 3025000000c = 171,160133 \\ 3850000a + 3025000000b + 2533300000000c = 143970,2593 \end{cases}$$

Получилось уравнение для графика:

$$y = 0,000415824 - 0,0000047479 * n + 0,0000000619 * n^2$$

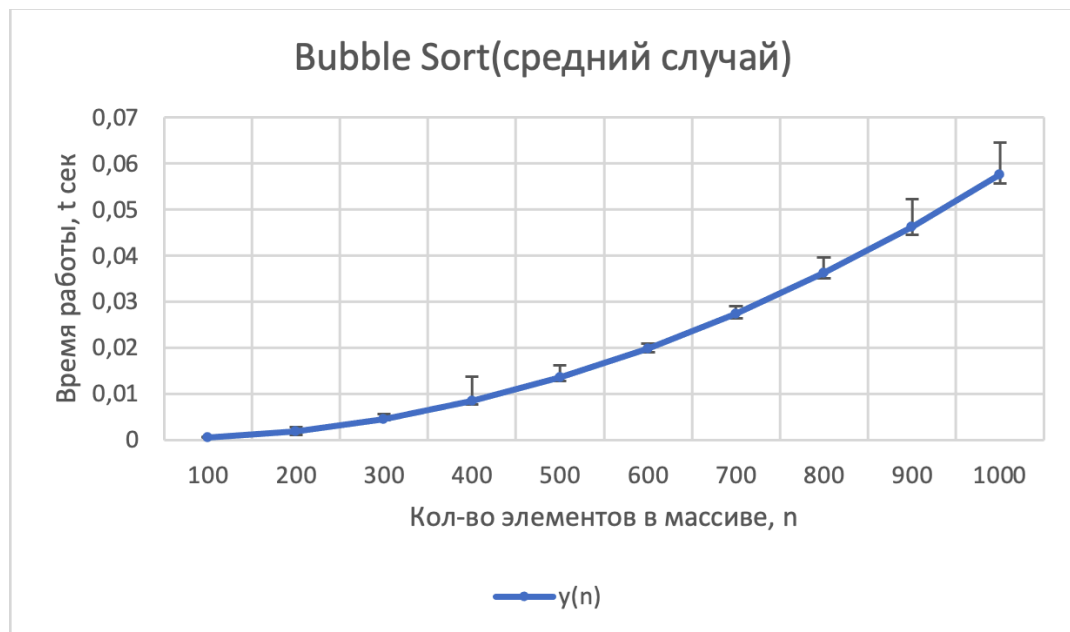


Рисунок 7. График с пределами погрешности

Из полученных результатов можно сделать вывод, что время работы программы зависит от количества элементов в массиве, то при росте длины массива увеличивается время работы - $O(n^2)$.