

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 №4
дисциплины «Программирование на Python»
Вариант

Выполнил:

Говоров Егор Юрьевич

2 курс, группа ИВТ-б-о-22-1,

09.03.02 «Информатика и

вычислительная техника»,

направленность (профиль)

«Программное обеспечение средств

вычислительной техники и

автоматизированных систем», очная

форма обучения

(подпись)

Руководитель практики:

Воронкин.Р.А, канд. технических наук,

доцент кафедры инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Ход работы:

1. Написал программу, которая подсчитывает время, затрачиваемое на выполнение алгоритма линейного поиска.

```
1  #include <iostream>
2  #include <ctime>
3  #include <cstdlib>
4  const int ARRAY_SIZE = 50;
5  int MaxSearch(int arr[], int n) {
6      int max = arr[0];
7      for (int i = 1; i < n; i++) {
8          if (arr[i] > max) {
9              max = arr[i];
10         }
11     }
12     return max;
13 }
14
15 int MinSearch(int arr[], int n) {
16     int min = arr[0];
17     for (int i = 1; i < n; i++) {
18         if (arr[i] < min) {
19             min = arr[i];
20         }
21     }
22     return min;
23 }
24 int main() {
25     srand(static_cast<unsigned>(time(0)));
26     int sizes[ARRAY_SIZE];
27     for (int i = 0; i < ARRAY_SIZE; i++) {
28         sizes[i] = 100 + i * 60;
29     }
30     std::cout << "Array Size\tMax Time (ms)\tMax Value\tMin Time (ms)\tMin Value\n";
31     for (int i = 0; i < ARRAY_SIZE; i++) {
32         int arrSize = sizes[i];
33         int* arr = new int[arrSize];
34
35         for (int j = 0; j < arrSize; j++) {
36             arr[j] = rand() % 1000;
37         }
38         int max, min;
39         clock_t start, end;
40         double MaxTime = 0, MinTime = 0;
41         int numIterations = 50;
42         for (int j = 0; j < numIterations; j++) {
43             start = clock();
44             max = MaxSearch(arr, arrSize);
45             end = clock();
46             MaxTime += (double)(end - start) / CLOCKS_PER_SEC * 1000;
47
48             start = clock();
49             min = MinSearch(arr, arrSize);
50             end = clock();
51             MinTime += (double)(end - start) / CLOCKS_PER_SEC * 1000;
52         }
53         MaxTime /= numIterations;
54         MinTime /= numIterations;
55         std::cout << arrSize << "\t" << MaxTime << "\t" << max << "\t" << MinTime << "\t" << min << "\n";
56         delete[] arr;
57     }
58     return 0;
59 }
60
```

Рис 1. Выполнение программы

Array Size	Max Time (ms)	Max Value	Min Time (ms)	Min Value
100	0.00152 990	0.00156 15		
160	0.0017 997	0.00168 21		
220	0.00186 998	0.00188 1		
280	0.00208 996	0.00236 2		
340	0.00228 997	0.0024 4		
400	0.00274 994	0.00264 4		
460	0.0026 999	0.00262 0		
520	0.00326 998	0.00278 0		
580	0.00294 996	0.00284 4		
640	0.00306 999	0.0031 1		
700	0.00332 998	0.00348 0		
760	0.00348 997	0.0034 1		
820	0.00388 999	0.00426 0		
880	0.00406 999	0.00378 0		
940	0.00402 998	0.00396 0		
1000	0.00422 999	0.00464 1		
1060	0.00432 999	0.00446 0		
1120	0.0048 997	0.00444 1		
1180	0.00498 999	0.00468 0		
1240	0.00494 999	0.00498 0		
1300	0.00514 999	0.00516 1		
1360	0.00576 997	0.00528 1		
1420	0.00546 999	0.00552 0		
1480	0.00546 998	0.00564 0		
1540	0.00588 999	0.00578 0		
1600	0.00614 998	0.00598 0		
1660	0.00696 999	0.00612 0		
1720	0.00664 999	0.00636 0		
1780	0.00652 999	0.00652 0		
1840	0.00682 999	0.00692 0		
1900	0.00708 999	0.0068 0		
1960	0.00738 999	0.00716 0		
2020	0.00726 999	0.00736 0		
2080	0.0076 999	0.00756 0		
2140	0.00786 999	0.00772 0		
2200	0.0083 999	0.0081 0		
2260	0.00808 999	0.00824 0		
2320	0.00818 999	0.00828 0		
2380	0.00856 999	0.00846 0		
2440	0.00846 999	0.00864 0		
2500	0.00884 999	0.00894 0		
2560	0.00884 999	0.00904 0		
2620	0.00976 999	0.00914 0		
2680	0.0093 999	0.00922 0		
2740	0.00972 999	0.00992 0		
2800	0.01064 999	0.01048 0		
2860	0.01112 999	0.01074 0		
2920	0.01104 999	0.0108 0		
2980	0.0112 999	0.01116 0		
3040	0.01148 999	0.01204 0		

Рис 2. Итоговый вывод времени и значений

2. Перенес данные по алгоритму поиска максимума в таблицу Excel и произвел необходимые расчеты

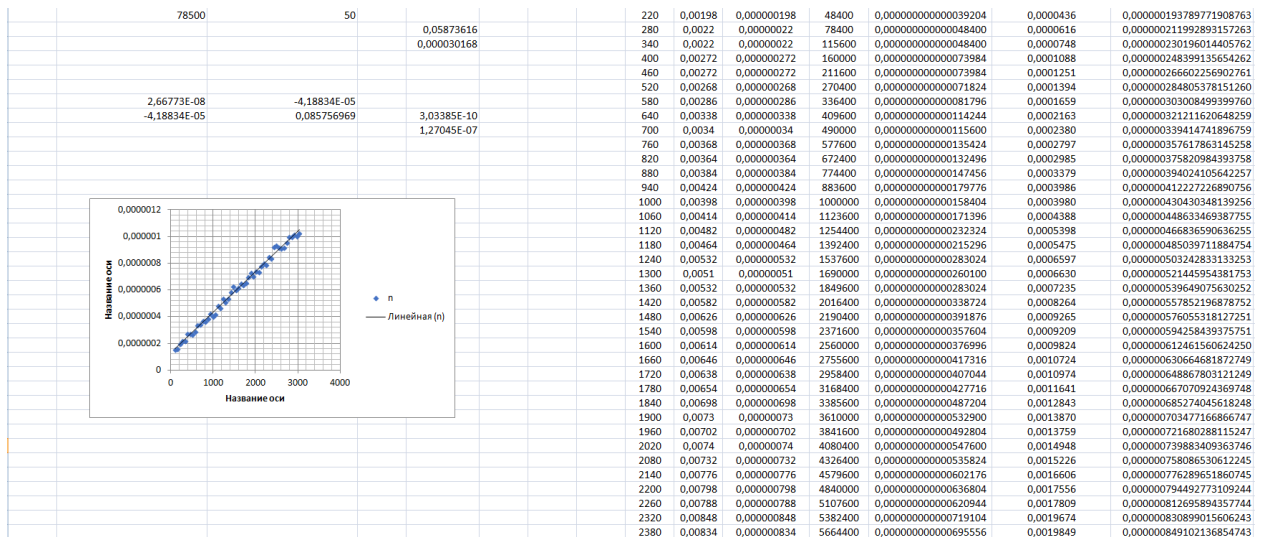


Рисунок 2. Расчет линейной зависимости

3. Построил график линейной зависимости времени выполнения алгоритма поиска максимума в массиве от размера массива

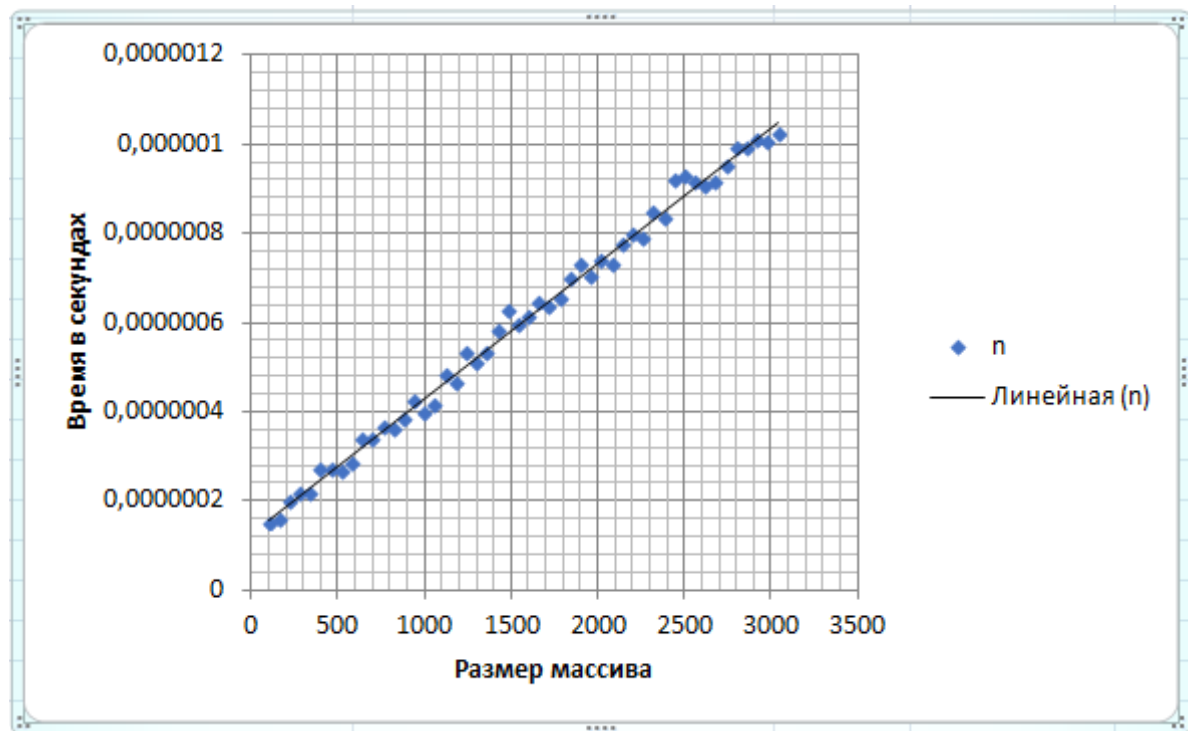


Рисунок 3. График для поиска максимума

4. Произвел аналогичные расчеты для получения необходимой функции.

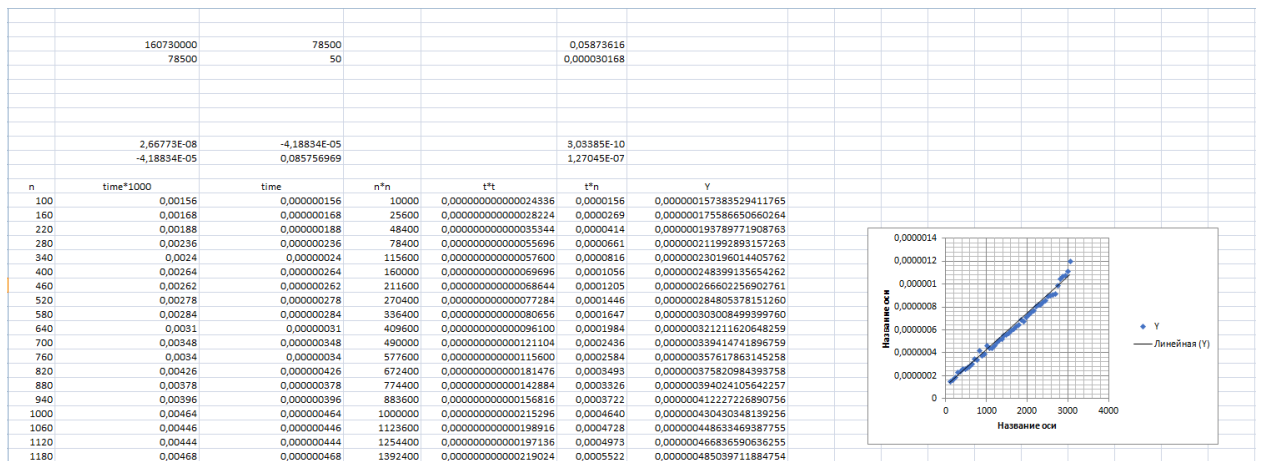


Рисунок 4. Расчет функции линейной зависимости для поиска минимума

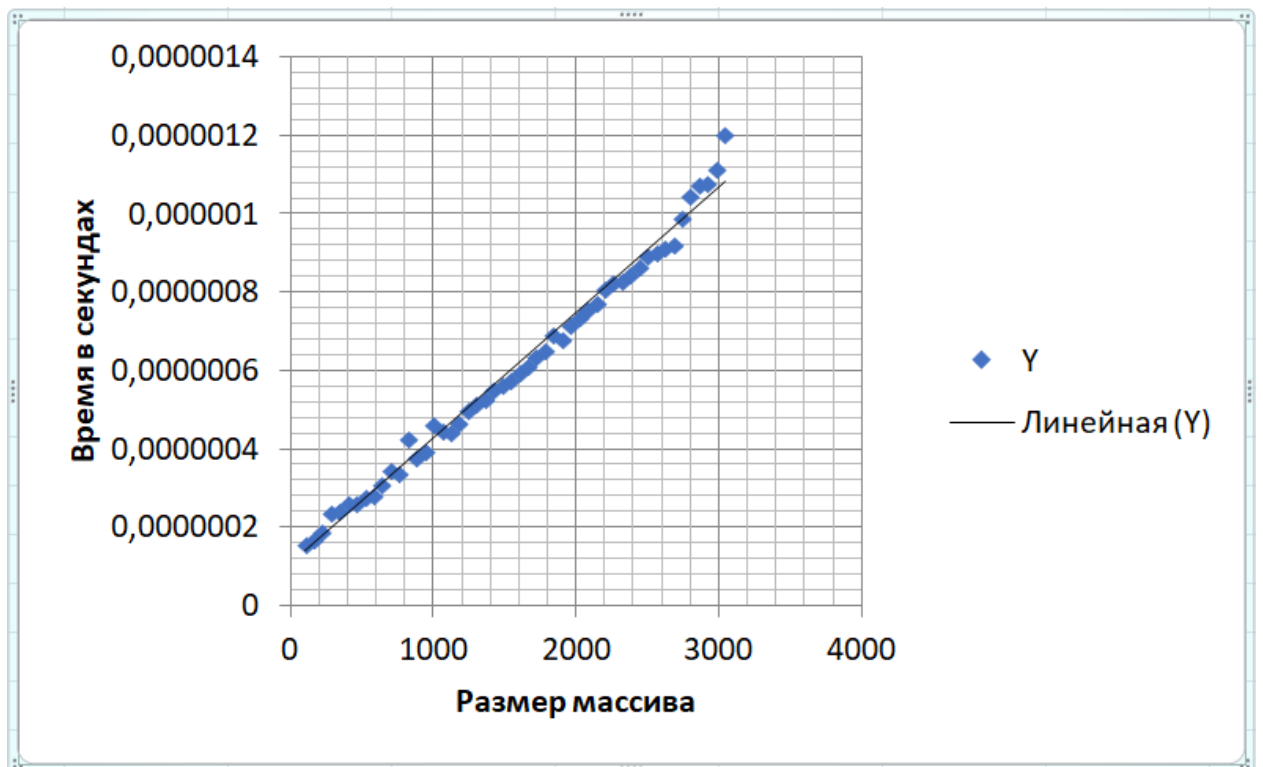


Рисунок 4. График для поиска минимума

5. Рассчитал коэффициенты парной корреляции для поиска максимума ($r = 0,997057873$) и минимума ($r = 0,994173368$).

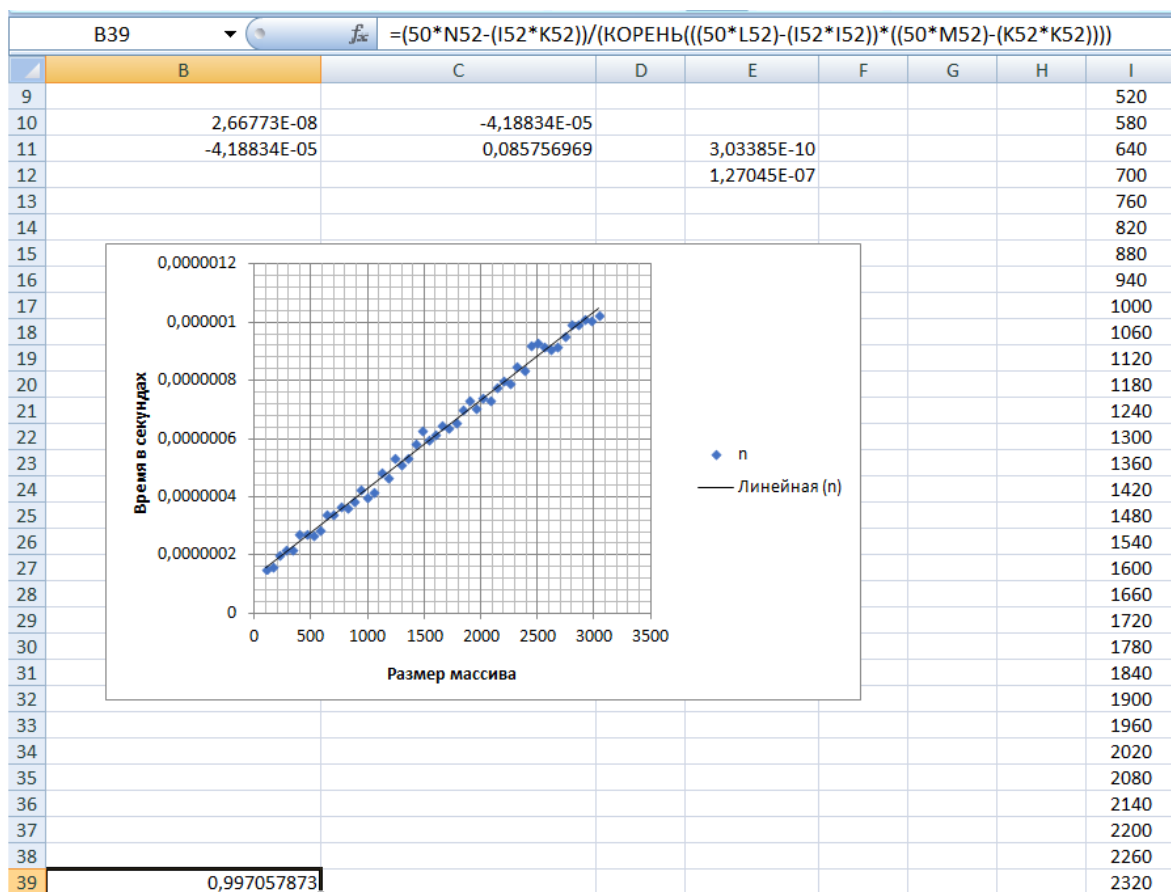


Рисунок 7. Расчет коэффициента парной корреляции для поиска максимума

B68 $f_x = \frac{(50*F64-(A64*C64))}{(КОРЕНЬ(((50*D64)-(A64*A64))*((50*E64)-(C64*C64))))}$

	B	C	D	E	F
39	0,00598	0,000000598	2560000	0,000000000000357604	0,0009568
40	0,00612	0,000000612	2755600	0,000000000000374544	0,0010159
41	0,00636	0,000000636	2958400	0,000000000000404496	0,0010939
42	0,00652	0,000000652	3168400	0,000000000000425104	0,0011606
43	0,00692	0,000000692	3385600	0,000000000000478864	0,0012733
44	0,0068	0,00000068	3610000	0,000000000000462400	0,0012920
45	0,00716	0,000000716	3841600	0,000000000000512656	0,0014034
46	0,00736	0,000000736	4080400	0,000000000000541696	0,0014867
47	0,00756	0,000000756	4326400	0,000000000000571536	0,0015725
48	0,00772	0,000000772	4579600	0,000000000000595984	0,0016521
49	0,0081	0,00000081	4840000	0,000000000000656100	0,0017820
50	0,00824	0,000000824	5107600	0,000000000000678976	0,0018622
51	0,00828	0,000000828	5382400	0,000000000000685584	0,0019210
52	0,00846	0,000000846	5664400	0,000000000000715716	0,0020135
53	0,00864	0,000000864	5953600	0,000000000000746496	0,0021082
54	0,00894	0,000000894	6250000	0,000000000000799236	0,0022350
55	0,00904	0,000000904	6553600	0,000000000000817216	0,0023142
56	0,00914	0,000000914	6864400	0,000000000000835396	0,0023947
57	0,00922	0,000000922	7182400	0,000000000000850084	0,0024710
58	0,00992	0,000000992	7507600	0,000000000000984064	0,0027181
59	0,01048	0,000001048	7840000	0,000000000001098304	0,0029344
60	0,01074	0,000001074	8179600	0,000000000001153476	0,0030716
61	0,0108	0,00000108	8526400	0,000000000001166400	0,0031536
62	0,01116	0,000001116	8880400	0,000000000001245456	0,0033257
63	0,01204	0,000001204	9241600	0,000000000001449616	0,0036602
64	0,30502	0,000030502	160730000	0,000000000022498764	0,0598953
65					
66					
67					
68		0,994173368			

Рисунок 8. Расчет коэффициента парной корреляции для поиска минимума

Вывод: В процессе выполнения лабораторной работы был проведен анализ работы алгоритмов по поиску максимума и минимума в массивах разных размеров. Были сделаны выводы о том, что время затрачиваемое на перебор членов массива зависит от его размера, что было подтверждено с помощью экспериментально. Благодаря полученным результатам можно сделать вывод, что данные линейно зависимы от размеров массива.