

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 №3
дисциплины «Алгоритмизация»
Вариант 7

Выполнил:
Горбунов Данила Евгеньевич
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. 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, предусмотрел варианты среднего (искомый элемент находится где-то в середине массива) и худшего (искомый элемент не найден) случая.

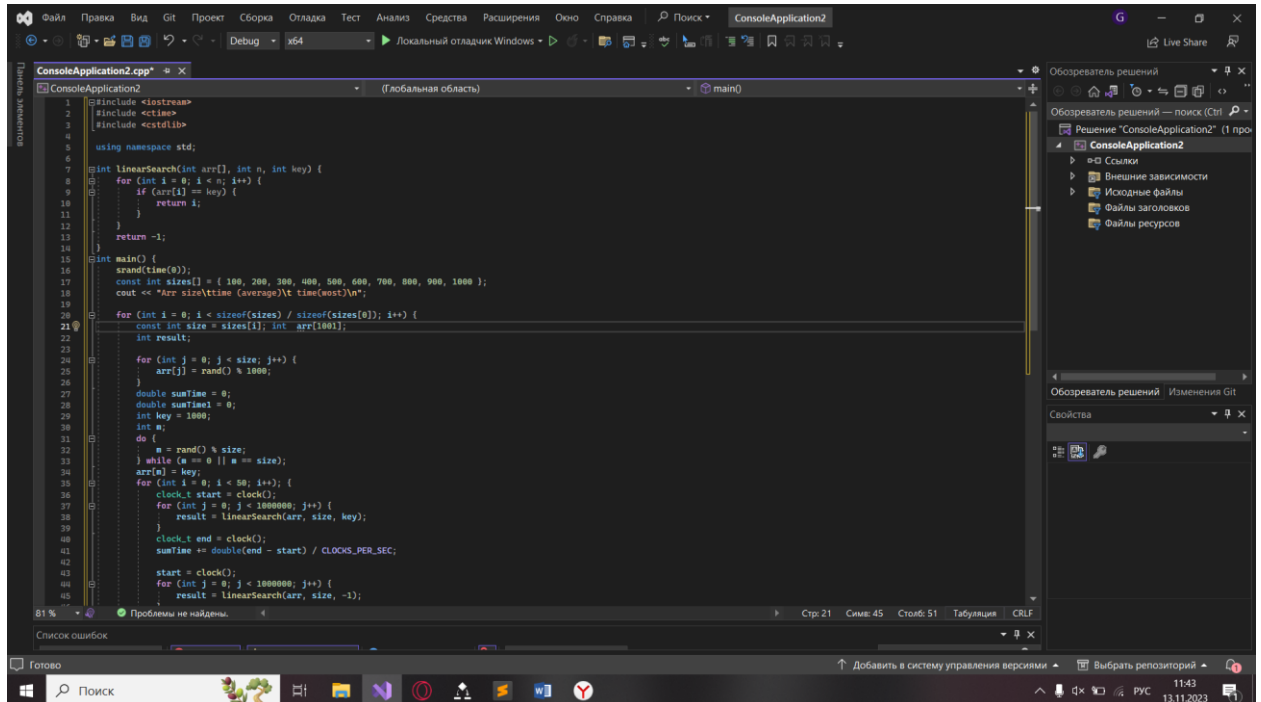


Рисунок 1. Программа

```
#include <iostream>
#include <ctime>
#include <cstdlib>
```

```
using namespace std;
```

```
int linearSearch(int arr[], int n, int key) {
    for (int i = 0; i < n; i++) {
        if (arr[i] == key) {
            return i;
        }
    }
    return -1;
}
```

```
int main() {
    srand(time(0));
    const int sizes[] = { 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000 };
    cout << "Arr size\ttime (average)\ttime (wost)\n";
```

```
    for (int i = 0; i < sizeof(sizes) / sizeof(sizes[0]); i++) {
        const int size = sizes[i]; int arr[1001];
        int result;
```

```
        for (int j = 0; j < size; j++) {
            arr[j] = rand() % 1000;
        }
        double sumTime = 0;
        double sumTime1 = 0;
```

```

int key = 1000;
int m;
do {
    m = rand() % size;
} while (m == 0 || m == size);
arr[m] = key;
for (int i = 0; i < 50; i++) {
    clock_t start = clock();
    for (int j = 0; j < 1000000; j++) {
        result = linearSearch(arr, size, key);
    }
    clock_t end = clock();
    sumTime += double(end - start) / CLOCKS_PER_SEC;

    start = clock();
    for (int j = 0; j < 1000000; j++) {
        result = linearSearch(arr, size, -1);
    }
    end = clock();
    sumTime1 += double(end - start) / CLOCKS_PER_SEC;
}
cout << size << "\t\t" << sumTime / 50 << "sec\t\t" << sumTime1 / 50 << "sec\n";
}
return 0;
}

```

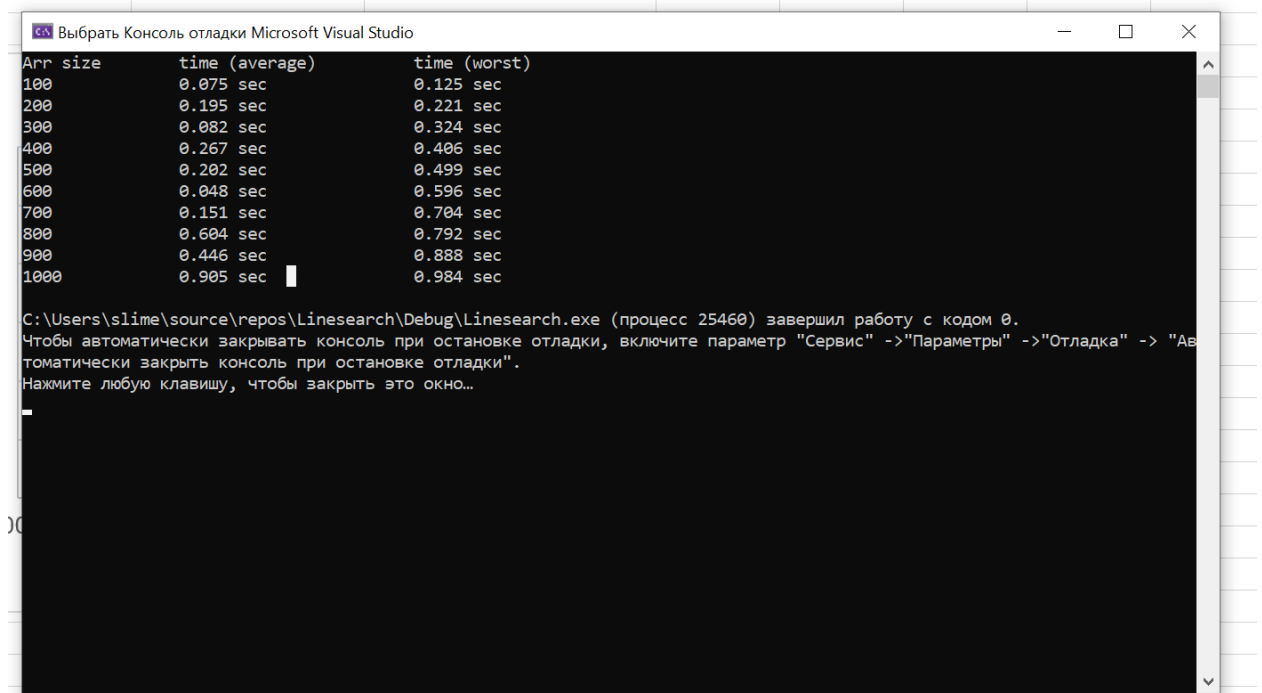


Рисунок 2. Результат работы программы

Таблица 1. Время работы алгоритма линейного поиска

Размер массива (n)	Средний случай (сек * 100000)	Худший случай (сек * 100000)
100	0,075	0,125
200	0,195	0,221
300	0,082	0,324
400	0,267	0,406
500	0,202	0,499

600	0,048	0,596
700	0,151	0,704
800	0,604	0,792
900	0,446	0,888
1000	0,905	0,984

2. Перенес данные в таблицу Excel и произвел необходимые расчеты для метода наименьших квадратов.

	A	B	C	D	E	F	G	H
1		n	time*10000	time	n*n	time*time	time*n	Y
2		100	0,075	0,0000075000	10000	0,0000000000562500	0,00075000	0,0000022411
3		200	0,195	0,0000195000	40000	0,0000000003802500	0,00390000	0,0000083742
4		300	0,082	0,0000082000	90000	0,0000000000672400	0,00246000	0,0000145073
5		400	0,276	0,0000276000	160000	0,00000000007617600	0,01104000	0,0000206404
6		500	0,202	0,0000202000	250000	0,00000000004080400	0,01010000	0,0000267735
7		600	0,048	0,0000048000	360000	0,0000000000230400	0,00288000	0,0000329065
8		700	0,151	0,0000151000	490000	0,00000000002280100	0,01057000	0,0000390396
9		800	0,604	0,0000604000	640000	0,00000000036481600	0,04832000	0,0000451727
10		900	0,446	0,0000446000	810000	0,00000000019891600	0,04014000	0,0000513058
11		1000	0,905	0,0000905000	1000000	0,00000000081902500	0,09050000	0,0000574389
12	сумма	5500	2,984	0,0002984000	3850000	0,0000000157521600	0,22066000	
13								
14								
15	yp1	a*сумм(n*n)+b*сумм(n)=сумм(t*n)		385000*a+5500b=0,0022066				
16	yp2	a*сумм(n)+b*N=сумм(t)		5500a+10b=0,000002984				
17								
18								
19								
20		Матричный способ решения системы:						
21		385000	5500		0,0022066			
22		5500	10		0,0002984			
23								
24		-3,78788E-07	0,000208333	a=	6,13308E-08			
25		0,000208333	-0,014583333	b=	-3,89196E-06			

Рисунок 3. Расчет линейной зависимости

3. Построил график линейной зависимости времени выполнения линейного поиска от размера массива в среднем случае.

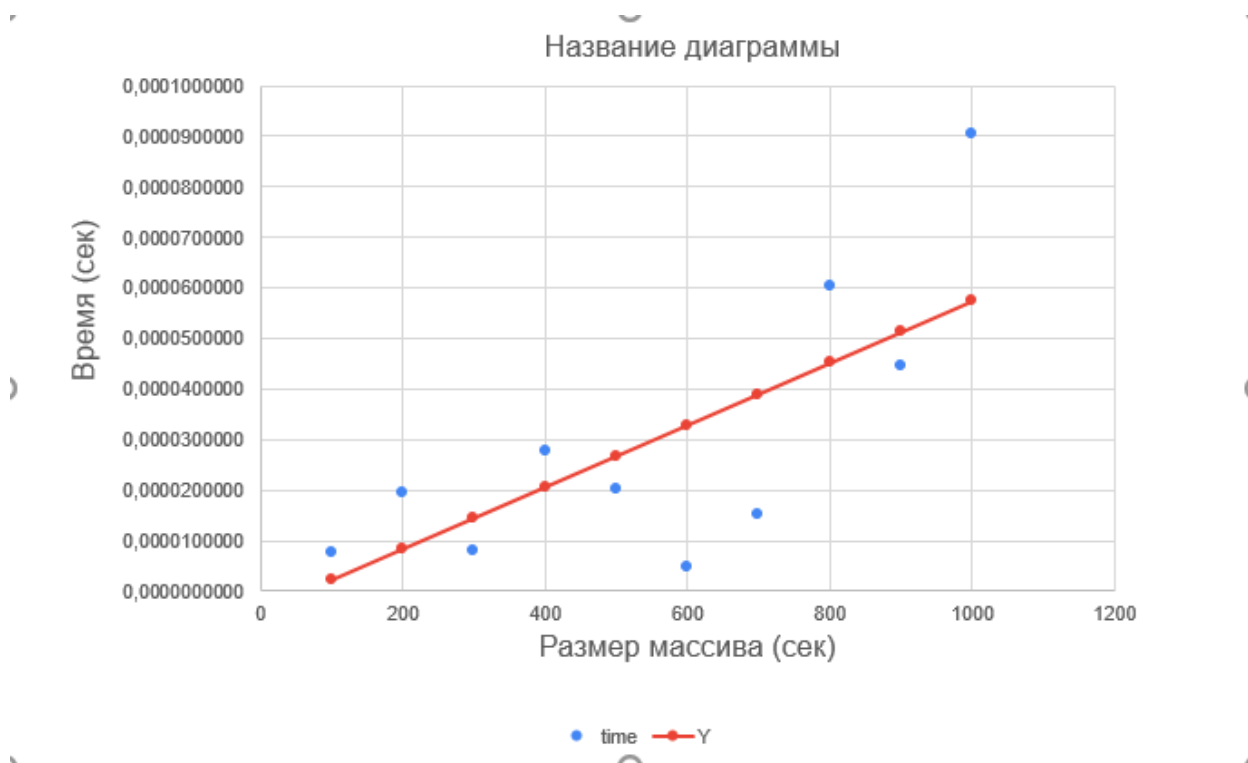


Рисунок 4. График для среднего случая

4. Произвел аналогичные расчеты для получения необходимой функции.

	A	B	C	D	E	F	G	H
1	n		time*10000	time	n*n	t*t	time*n	Y
2		100	0,125	0,0000125000	10000	0,00000000015625	0,00125	0,000009995397727
3		200	0,221	0,0000221000	40000	0,00000000048841	0,00442	0,000020083087121
4		300	0,324	0,0000324000	90000	0,00000000104976	0,00972	0,000030170776515
5		400	0,406	0,0000406000	160000	0,00000000164836	0,01624	0,000040258465909
6		500	0,499	0,0000499000	250000	0,00000000249001	0,02495	0,000050346155303
7		600	0,596	0,0000596000	360000	0,00000000355216	0,03576	0,000060433844697
8		700	0,704	0,0000704000	490000	0,00000000495616	0,04928	0,000070521534091
9		800	0,792	0,0000792000	640000	0,00000000627264	0,06336	0,000080609223485
10		900	0,888	0,0000888000	810000	0,00000000788544	0,07992	0,000090696912879
11		1000	0,984	0,0000984000	1000000	0,00000000968256	0,0984	0,000100784602273
12	сумма	5500	5,539	0,0005539000	3850000	0,00000003818175	0,3833	
13								

Рисунок 5. Расчет функции линейной зависимости для худшего случая

5. Построил график линейной зависимости времени выполнения линейного поиска от размера массива в худшем случае.

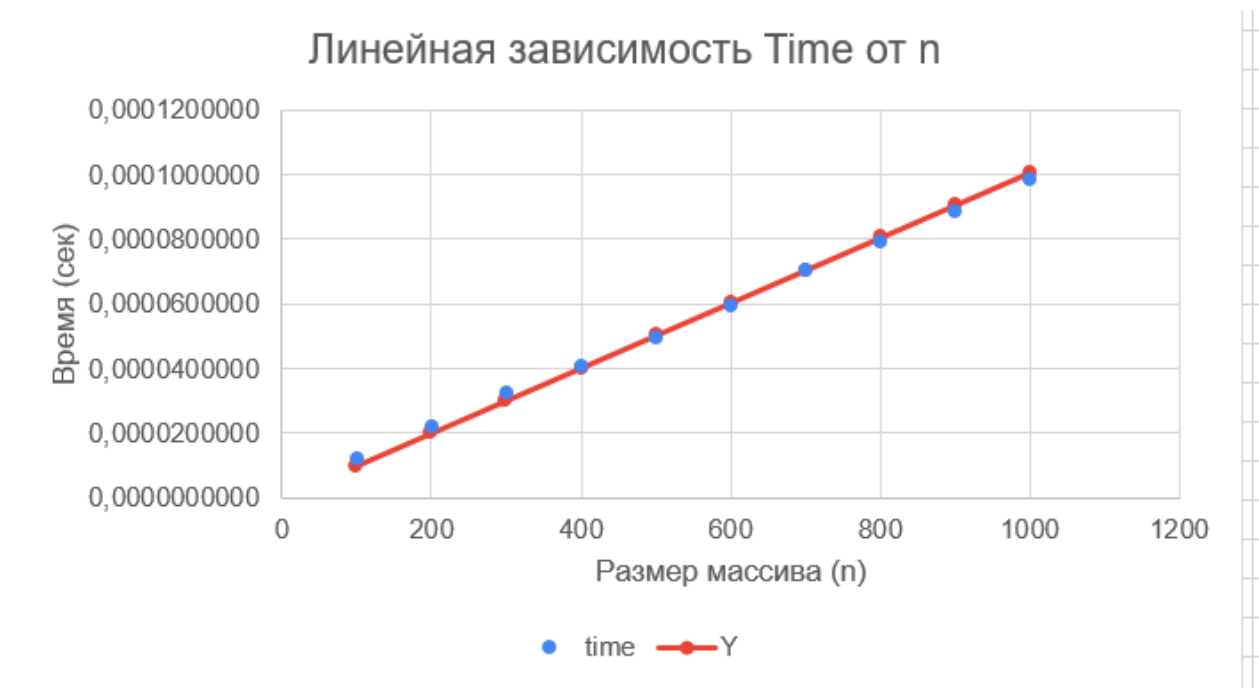


Рисунок 6. График для худшего случая случая

6. Рассчитал коэффициенты парной корреляции для общего ($r = 0.7522288$) и худшего ($r = 0.9998461$) случая.

B27 $\text{=КОРРЕЛ(B2:B11;D2:D11)}$								
A	B	C	D	E	F	G	H	
	n	time*10000	time	n*n	time*time	time*n	Y	
1								
2	100	0,075	0,0000075000	10000	0,0000000000562500	0,00075000	0,0000022411	
3	200	0,195	0,0000195000	40000	0,0000000003802500	0,00390000	0,0000083742	
4	300	0,382	0,0000082000	90000	0,000000000672400	0,00246000	0,0000145073	
5	400	0,276	0,0000276000	160000	0,0000000007617600	0,01104000	0,0000206404	
6	500	0,202	0,0000202000	250000	0,0000000004080400	0,01010000	0,0000267735	
7	600	0,048	0,0000048000	360000	0,0000000000230400	0,00288000	0,0000329065	
8	700	0,151	0,0000151000	490000	0,0000000002280100	0,01057000	0,0000390396	
9	800	0,604	0,0000604000	640000	0,00000000036481600	0,04832000	0,0000451727	
10	900	0,446	0,0000446000	810000	0,0000000019891600	0,04014000	0,0000513058	
11	1000	0,905	0,0000905000	1000000	0,0000000081902500	0,09050000	0,0000574389	
12	сумма	5500	2,984	3850000	0,000000157521600	0,22066000		
13								
14								
15	yp1	$a \cdot \text{сумм}(n \cdot n) + b \cdot \text{сумм}(n) = \text{сумм}(t \cdot n)$		$385000 \cdot a + 5500 \cdot b = 0,0022066$				
16	yp2	$a \cdot \text{сумм}(n) + b \cdot N = \text{сумм}(t)$		$5500 \cdot a + 10 \cdot b = 0,000002984$				
17								
18								
19								
20		Матричный способ решения системы:						
21		385000	5500		0,0022066			
22		5500	10		0,0002984			
23								
24		-3,78788E-07	0,000208333	a=	6,13308E-08			
25		0,000208333	-0,014583333	b=	-3,89196E-06			
26								
27	КОРРЕЛЯЦ	0,752228872						
28	КОРРЕЛЯЦ	0,752228872						
29								

Рисунок 7. Расчет коэффициента парной корреляции для общего случая

B17									
=((10*0,3833)-(5500*0,0005539))/(КОРЕНЬ((10*3850000-30250000)*(10*F12-(D12*D12))))									
	A	B	C	D	E	F	G	H	
1		n	time*10000	time	n*n	t*t	time*n	Y	
2		100	0,125	0,0000125000	10000	0,00000000015625	0,00125	0,000009995397727	
3		200	0,221	0,0000221000	40000	0,00000000048841	0,00442	0,000020083087121	
4		300	0,324	0,0000324000	90000	0,00000000104976	0,00972	0,000030170776515	
5		400	0,406	0,0000406000	160000	0,00000000164836	0,01624	0,000040258465909	
6		500	0,499	0,0000499000	250000	0,00000000249001	0,02495	0,000050346155303	
7		600	0,596	0,0000596000	360000	0,00000000355216	0,03576	0,000060433844697	
8		700	0,704	0,0000704000	490000	0,00000000495616	0,04928	0,000070521534091	
9		800	0,792	0,0000792000	640000	0,00000000627264	0,06336	0,000080609223485	
10		900	0,888	0,0000888000	810000	0,00000000788544	0,07992	0,000090696912879	
11		1000	0,984	0,0000984000	1000000	0,00000000968256	0,0984	0,000100784602273	
12	сумма	5500	5,539	0,0005539000	3850000	0,00000003818175	0,3833		
13									
14									
15									
16									
17	КОРЕЛЛ	0,999846082							
18									

Рисунок 8. Расчет коэффициента парной корреляции для худшего случая

Вывод: в ходе выполнения лабораторной работы был исследован алгоритм линейного поиска в массиве. Проведенный анализ позволяет утверждать, что время выполнения этого алгоритма в худшем и среднем случаях напрямую коррелирует с размером массива. Это утверждение подтверждено результатами экспериментов и статистическими методами, включая расчет коэффициента парной корреляции. Таким образом, можно сделать вывод о том, что этот алгоритм действительно обладает линейной зависимостью от размера массива, в котором выполняется поиск.