

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»  
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных  
систем

## **Лабораторная работа №4**

по дисциплине: Алгоритмы и структуры данных  
тема: «Сравнительный анализ алгоритмов поиска (С)»

Выполнил: ст. группы ПВ-202  
Буйвало Анастасия Андреевна

Проверил:  
Кабалянец Петр Степанович  
Маньшин Илья Михайлович

Белгород 2021 г.

## Лабораторная работа №4

### «Сравнительный анализ алгоритмов поиска (С)»

#### Цель работы:

Изучение алгоритмов поиска элемента в массиве и закрепление навыков в проведении сравнительного анализа алгоритмов.

#### Задание:

1. Изучить алгоритмы поиска:
  - 1) в неупорядоченном массиве:
    - линейный;
    - быстрый линейный;
  - 2) в упорядоченном массиве:
    - быстрый линейный;
    - бинарный;
    - блочный.
2. Разработать и программно реализовать средство для проведения экспериментов по определению временных характеристик алгоритмов поиска.
3. Провести эксперименты по определению временных характеристик алгоритмов поиска. Результаты экспериментов представить в виде таблиц 12 и 13. Клетки таблицы 12 содержат максимальное количество операций сравнения при выполнении алгоритма поиска, а клетки таблицы 13 — среднее число операций сравнения.
4. Построить графики зависимости количества операций сравнения от количества элементов в массиве.
5. Определить аналитическое выражение функции зависимости количества операций сравнения от количества элементов в массиве.
6. Определить порядок функций временной сложности алгоритмов поиска

## Выполнение работы:

### Задание 2:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

#define N 451

//линейный поиск
int lin_search(int a[], int n, int x, int* comps)
{
    int i = 0;
    while(i < n && ++(*comps) && a[i] != x){
        i++;
    }
    if(i == n)
        i = -1;
    return i;
}

//быстрый линейный поиск в неотсортированном массиве
int fast_n_lin_search(int a[], int n, int x, int* comps)
{
    int i = 0;
    a[n] = x;
    while(++(*comps) && a[i] != x){
        i++;
    }
    if(i == n)
        i = -1;
    return i;
}

//быстрый линейный поиск в отсортированном массиве
int fast_o_lin_search(int a[], int n, int x, int* comps)
{
    int i = 0;
    a[n] = x;
    while(++(*comps) && a[i] < x){
        i++;
    }
    if(++(*comps) && a[i] == x && i != n)
        i = i;
    else
        i = -1;
    return i;
}

//бинарный поиск
int bin_search(int a[], int n, int x, int *comps)
{
    int l = 0;
    int r = n - 1;
    int m;
    while(l <= r){
        m = (r + l)/2;
```

```

        if (++(*comps) && x == a[m])
            return m;
        if(++(*comps) && x > a[m])
            l = m + 1;
        else
            r = m - 1;
    }
    return -1;
}

int block_search(int a[], int n, int x, int *comps)
{
    int t = sqrt(n); //размера каждого блока
    int i = 0;
    while(i < n && ++(*comps) && a[i] < x) //блок в котором может быть x
        i+= t;
    int ti = i;
    if (i == 0) { //проверяем ситуацию, что x находится на 0 месте в массиве
        if (++(*comps) && a[i] == x) // x находился на 0 месте в массиве
            return 0;
        else //массив начинался с числа, которое больше x
            return -1;
    }
    for(; i >= ti - t; i--) { //поиск x в найденном блоке
        if (++(*comps) && a[i] == x) // в блоке найден x
            return i;
    }
    return -1;
}

void copy_arr(int a[], int b[], int n)
{
    for(int i = 0; i < n; i++)
        b[i] = a[i];
}

//0-по возрастанию, 1 - по убыванию, 2 - рандом
void generate_arr(int a[], int n, int t) {
    switch (t) {
        case 0: {
            for (int i = 0; i < n; i++)
                a[i] = i;
            break;
        }
        case 1: {
            int j = n - 1;
            for (int i = 0; i < n; i++) {
                a[i] = j;
                j--;
            }
            break;
        }
        case 2: {
            for (int i = 0; i < n; i++)
                a[i] = rand() % n;
            break;
        }
        default: {
            printf("передано неверное значение\n");
            break;
        }
    }
}

```

```

}

//все поиски для неотсортированного
void all_search_n(int a[])
{
    for(int i = 50; i <= 450; i += 50) {
        int n = i;
        int max1 = 0;
        int comps1 = 0;
        int max2 = 0;
        int comps2 = 0;
        int s1 = 0;
        int s2 = 0;
        for (int j = 0; j < 100; j++) { //100 тестов для определения максимального и
среднего
            int x = rand() % n ;
            generate_arr(a, n, 2);
            lin_search(a, n, x, &comps1);
            fast_n_lin_search(a, n, x, &comps2);
            s1 += comps1;
            s2 += comps2;
            if (comps1 > max1)
                max1 = comps1;
            if(comps2 > max2)
                max2 = comps2;
            comps1 = 0;
            comps2 = 0;
        }
        printf("максимальное и среднее число сравнений в линейном поиске(n = %d) :
max = %d, mid = %d\n", n, max1, s1/100);
        printf("максимальное и среднее число сравнений в быстром линейном поиске(не
отсорт)(n = %d) : max = %d, mid = %d\n", n, max2, s2/100);
    }
}

//все поиски для отсортированного
void all_search_o(int a[])
{
    srand(595);
    for(int i = 50; i <= 450; i += 50) {
        int max1 = 0;
        int comps1 = 0;
        int max2 = 0;
        int comps2 = 0;
        int max3 = 0;
        int comps3 = 0;
        int s1 = 0;
        int s2 = 0;
        int s3 = 0;
        int n = i;
        for ( int j = 1; j <= 100; j++) { //100 тестов для определения максимального
и среднего
            int x = rand() % n ;
            generate_arr(a, i, 0);
            if (j == 0) {
                bin_search(a, n, n - 1, &comps1);
                fast_o_lin_search(a, n, n-1, &comps2);
                block_search(a, n, n - 1, &comps3);
            }
            else {
                bin_search(a, n, x, &comps1);
                fast_o_lin_search(a, n, x, &comps2);
            }
        }
    }
}

```

```

        block_search(a, n, x, &comps3);
    }
    s1 += comps1;
    s2 += comps2;
    s3 += comps3;
    if (comps1 > max1)
        max1 = comps1;
    if (comps2 > max2)
        max2 = comps2;
    if (comps3 > max3)
        max3 = comps3;
    comps1 = 0;
    comps2 = 0;
    comps3 = 0;
}
    printf("максимальное и среднее число сравнений в бинарном поиске(n = %d) :
max = %d, mid = %d\n", i, max1, s1/100);
    printf("максимальное и среднее число сравнений в быстром линейном
поиске(отсорт)(n = %d) : max = %d, mid = %d\n", i, max2, s2/100);
    printf("максимальное и среднее число сравнений в блочном поиске(n = %d) : max
= %d, mid = %d\n", i, max3, s3/100);
}
}

void condit_search(int a[])
{
    all_search_n(a);
    all_search_o(a);
}

int main()
{
    srand(time(0));
    int a[N];
    condit_search(a);
    return 0;
}

```

### Задание 3:

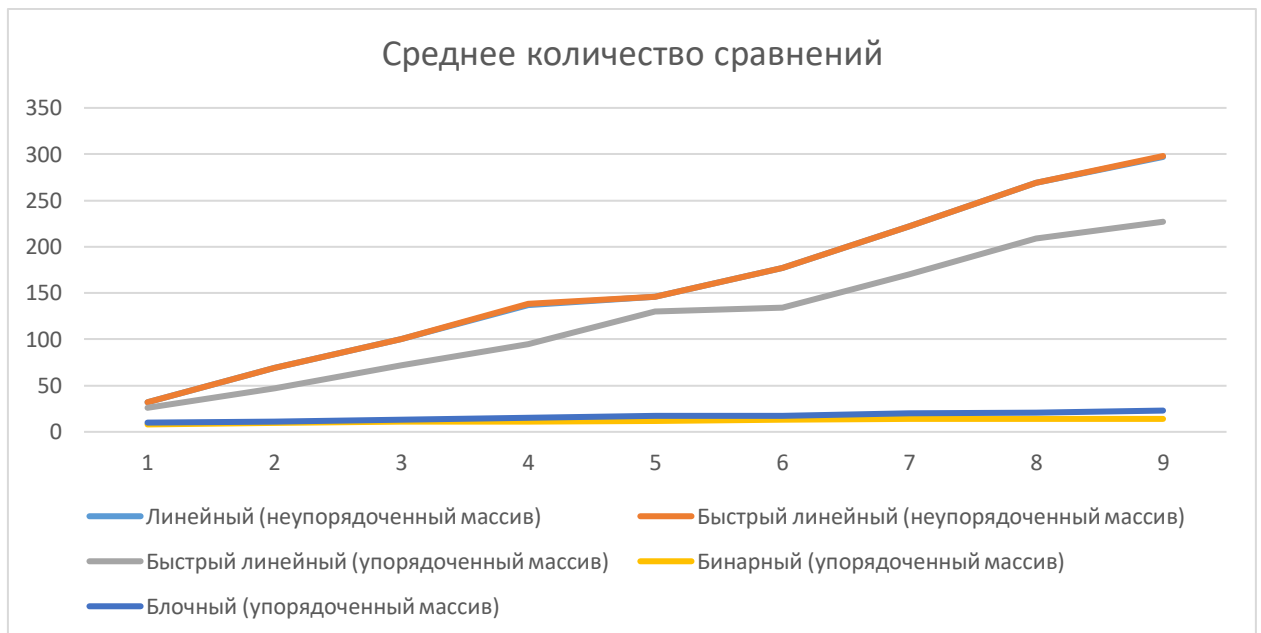
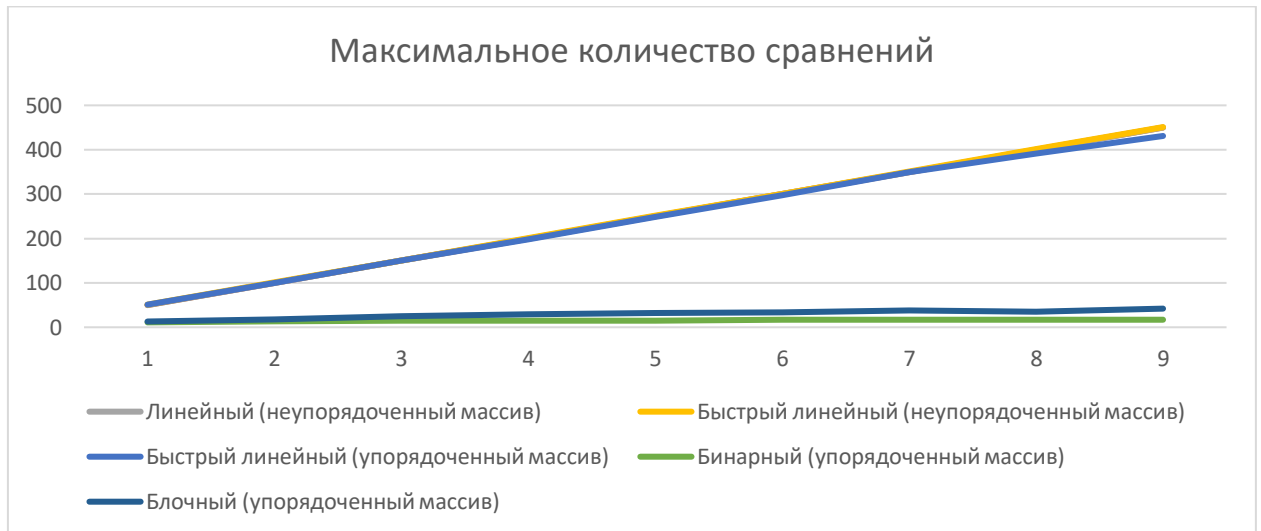
Максимальное кол-во операций сравнения:

Сортировка	Количество элементов в массиве								
	50	100	150	200	250	300	350	400	450
Линейный (неупорядоченный массив)	50	100	150	200	250	300	350	400	450
Быстрый линейный (неупорядоченный массив)	51	101	151	201	251	301	351	401	451
Быстрый линейный (упорядоченный массив)	51	100	151	198	249	298	349	392	431
Бинарный (упорядоченный массив)	11	13	15	15	15	17	17	17	17
Блочный (упорядоченный массив)	13	18	25	29	32	33	38	35	42

Среднее кол-во операций сравнения:

Сортировка	Количество элементов в массиве								
	50	100	150	200	250	300	350	400	450
Линейный (неупорядоченный массив)	32	69	100	137	146	177	222	269	297
Быстрый линейный (неупорядоченный массив)	32	69	100	138	146	177	222	269	298
Быстрый линейный (упорядоченный массив)	26	47	72	95	130	134	170	209	227
Бинарный (упорядоченный массив)	8	10	11	11	12	13	14	14	14
Блочный (упорядоченный массив)	10	11	13	15	17	17	20	21	23

#### Задание 4:



#### Задание 5:

- Линейный –  $N$  (без учета проверки на границы),  $N*2$  (с учетом проверки на границы)
- Линейный быстрый –  $N$
- Бинарный –  $\log_2(N)$
- Блочный –  $2 * \sqrt{N}$

#### Задание 6:

- Линейный –  $O(N)$
- Линейный быстрый –  $O(N)$
- Бинарный –  $O(\log_2(N))$



- Блочный –  $O(\sqrt{N})$

## Вывод:

Я изучила алгоритмы поиска элемента в массиве, реализовала их и выполнила эксперимент по определению количества операций сравнения при поиске. Приобрела навыки проведения сравнительного анализа различных алгоритмов поиска во время рассмотрения результатов эксперимента.

```

максимальное и среднее число сравнений в блочном поиске(n = 150) : max = 25, mid = 13
максимальное и среднее число сравнений в бинарном поиске(n = 200) : max = 15, mid = 11
максимальное и среднее число сравнений в быстром линейном поиске(отсорт)(n = 200) : max = 198, mid = 95
максимальное и среднее число сравнений в блочном поиске(n = 200) : max = 29, mid = 15
максимальное и среднее число сравнений в бинарном поиске(n = 250) : max = 15, mid = 12
максимальное и среднее число сравнений в быстром линейном поиске(отсорт)(n = 250) : max = 249, mid = 130
максимальное и среднее число сравнений в блочном поиске(n = 250) : max = 32, mid = 17
максимальное и среднее число сравнений в бинарном поиске(n = 300) : max = 17, mid = 13
максимальное и среднее число сравнений в быстром линейном поиске(отсорт)(n = 300) : max = 298, mid = 134
максимальное и среднее число сравнений в блочном поиске(n = 300) : max = 33, mid = 17
максимальное и среднее число сравнений в бинарном поиске(n = 350) : max = 17, mid = 14
максимальное и среднее число сравнений в быстром линейном поиске(отсорт)(n = 350) : max = 349, mid = 170
максимальное и среднее число сравнений в блочном поиске(n = 350) : max = 38, mid = 20
максимальное и среднее число сравнений в бинарном поиске(n = 400) : max = 17, mid = 14
максимальное и среднее число сравнений в быстром линейном поиске(отсорт)(n = 400) : max = 392, mid = 209
максимальное и среднее число сравнений в блочном поиске(n = 400) : max = 35, mid = 21
максимальное и среднее число сравнений в бинарном поиске(n = 450) : max = 17, mid = 14
максимальное и среднее число сравнений в быстром линейном поиске(отсорт)(n = 450) : max = 431, mid = 227
максимальное и среднее число сравнений в блочном поиске(n = 450) : max = 42, mid = 23

```

```

/Users/nastabujvalo/CLionProjects/AiSD4/cmake-build-debug/AiSD4

```

```

максимальное и среднее число сравнений в линейном поиске(n = 50) : max = 50, mid = 32
максимальное и среднее число сравнений в быстром линейном поиске(не отсорт)(n = 50) : max = 51, mid = 32
максимальное и среднее число сравнений в линейном поиске(n = 100) : max = 100, mid = 69
максимальное и среднее число сравнений в быстром линейном поиске(не отсорт)(n = 100) : max = 101, mid = 69
максимальное и среднее число сравнений в линейном поиске(n = 150) : max = 150, mid = 100
максимальное и среднее число сравнений в быстром линейном поиске(не отсорт)(n = 150) : max = 151, mid = 100
максимальное и среднее число сравнений в линейном поиске(n = 200) : max = 200, mid = 137
максимальное и среднее число сравнений в быстром линейном поиске(не отсорт)(n = 200) : max = 201, mid = 138
максимальное и среднее число сравнений в линейном поиске(n = 250) : max = 250, mid = 146
максимальное и среднее число сравнений в быстром линейном поиске(не отсорт)(n = 250) : max = 251, mid = 146
максимальное и среднее число сравнений в линейном поиске(n = 300) : max = 300, mid = 177
максимальное и среднее число сравнений в быстром линейном поиске(не отсорт)(n = 300) : max = 301, mid = 177
максимальное и среднее число сравнений в линейном поиске(n = 350) : max = 350, mid = 222
максимальное и среднее число сравнений в быстром линейном поиске(не отсорт)(n = 350) : max = 351, mid = 222
максимальное и среднее число сравнений в линейном поиске(n = 400) : max = 400, mid = 269
максимальное и среднее число сравнений в быстром линейном поиске(не отсорт)(n = 400) : max = 401, mid = 269
максимальное и среднее число сравнений в линейном поиске(n = 450) : max = 450, mid = 297
максимальное и среднее число сравнений в быстром линейном поиске(не отсорт)(n = 450) : max = 451, mid = 298
максимальное и среднее число сравнений в бинарном поиске(n = 50) : max = 11, mid = 8
максимальное и среднее число сравнений в быстром линейном поиске(отсорт)(n = 50) : max = 51, mid = 26
максимальное и среднее число сравнений в блочном поиске(n = 50) : max = 15, mid = 8
максимальное и среднее число сравнений в бинарном поиске(n = 100) : max = 13, mid = 10
максимальное и среднее число сравнений в быстром линейном поиске(отсорт)(n = 100) : max = 100, mid = 47
максимальное и среднее число сравнений в блочном поиске(n = 100) : max = 18, mid = 11
максимальное и среднее число сравнений в бинарном поиске(n = 150) : max = 15, mid = 11
максимальное и среднее число сравнений в быстром линейном поиске(отсорт)(n = 150) : max = 151, mid = 72

```