

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Пермский национальный исследовательский политехнический университет»
(ПНИПУ)
Электротехнический факультет
Кафедра «Информационные технологии и автоматизированные системы»
(ИТАС)

Лабораторная работа

На тему
«Поиски»

Выполнил
Студент группы ИВТ-23-16
Адаев Даниил Дмитриевич
Проверил
Доцент кафедры ИТАС
Яруллин Д. В.

г. Пермь, 2024

Задание

В массиве целых чисел найти любой элемент с заданным значением.

Заполнение массива организовать любым удобным способом. При этом учесть, что не все поиски работают с неотсортированными массивами.

Поиск выполнить тремя видами:

1. Линейный;
2. Бинарный;
3. Интерполяционный.

Алгоритмы поисков

Линейный поиск идет с начала массива и до тех пор, пока не встретит элемент с нужным значением, после этого он выдает индекс, на котором был найден элемент. Если элемента не было в массиве, будет выдана соответствующая надпись.

Бинарный поиск работает только с отсортированным массивом. Он ищет середину массива, и смотрит, равен ли этот элемент искомому, если нет, то проверяет, в левой или правой части находится нужный элемент, после этого границы массива становятся границами нужной половины, далее в этих границах снова ищется середина, если элемент снова не найден, снова берется нужная половина, и т.д. пока не найдем элемент. Если же элемент не найден и правая граница равна левой, значит в массиве нет элемента с таким значением и нужно выдать соответствующую надпись.

Интерполяционный поиск работает точно также, как и бинарный, но для вычисления середины используется формула: $mid = l + ((searchNum - arr[l]) * (r - l)) / (arr[r] - arr[l])$, где l и r – правая и левая граница, $searchNum$ – значение искомого элемента, arr – массив, в котором производится поиск.

Линейный поиск

Код на C++

```
#include <iostream>
using namespace std;
struct arr
```

```

{
    int* a;
    int size;
};

int* createArray(int size, int max, int min)
{
    srand(time(0));
    int* a = new int[size];

    for (int i = 0; i < size; i++)
    {
        a[i] = rand() % max + min;

        cout << a[i] << " ";
    }

    cout << endl << endl;

    return a;
}

void linearSearch(arr a)
{
    int key;
    bool f = false;

    cout << "Enter key" << endl;
    cin >> key;

    for (int i = 0; i < a.size; i++)
    {
        if (a.a[i] == key)
        {
            f = true;
            cout << i + 1 << " ";
        }
    }

    if (f == false)
    {
        cout << "Element not found";
    }

    cout << endl << "!!!!!! !!!!! !!!!! !!!!! !!!!!" << endl;

    linearSearch(a);
}

int main()
{
    int n = 0, max = -1, min = -1, key;
    arr a;

    while (n <= 0)
    {
        cout << "Enter number of elements" << endl;
        cin >> n;
    }

    while (max < 0)
    {
        cout << "Enter maximum value (>= 0)" << endl;
        cin >> max;
    }
}

```

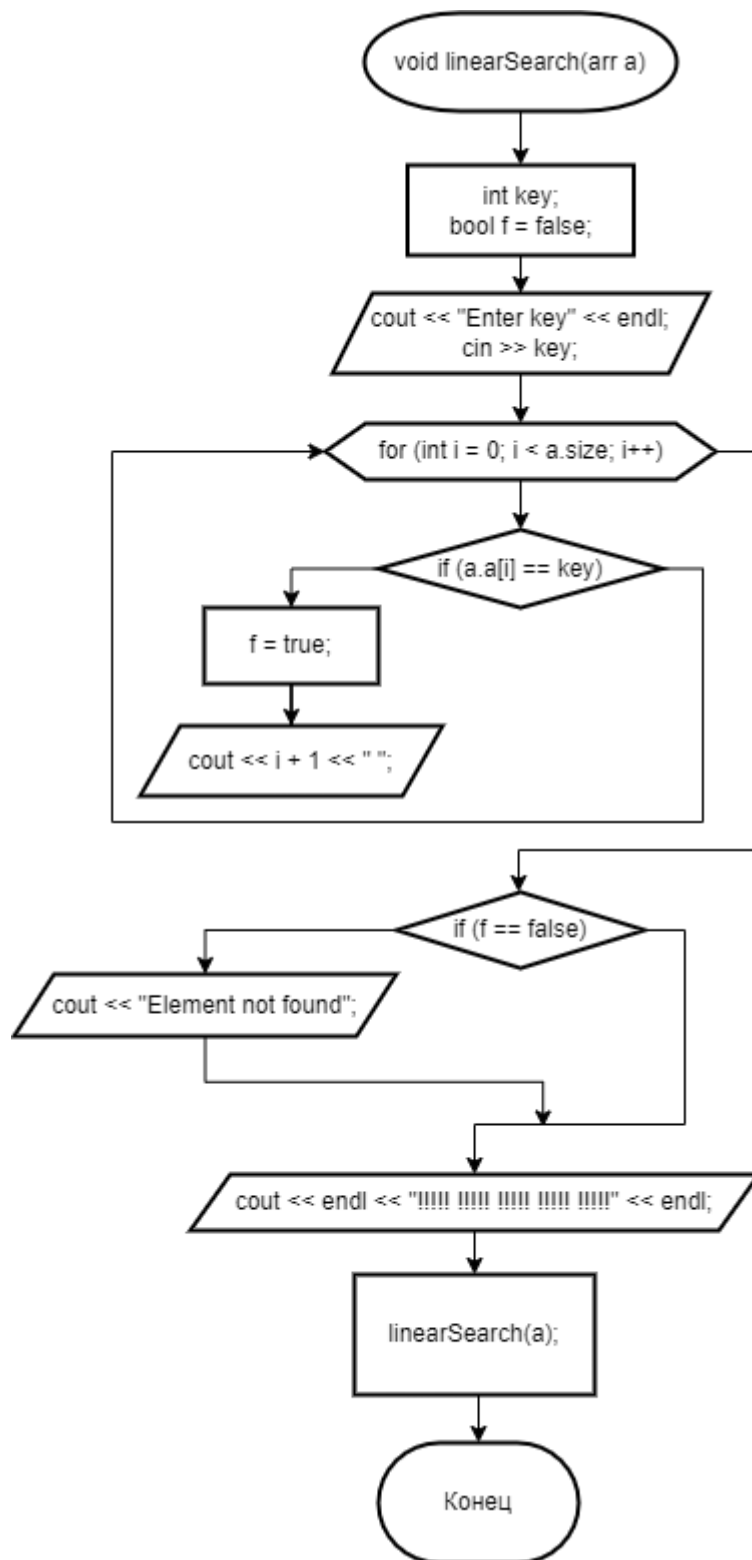
```
while (min < 0)
{
    cout << "Enter minimum value (>= 0)" << endl;
    cin >> min;
}

a.a = createArray(n, max, min);
a.size = n;

linearSearch(a);

return 0;
}
```

Блок-схема



Бинарный поиск

Код на C++

```

#include <iostream>

using namespace std;

struct arr
{
    int* a;

```

```

        int size;
    };

    int* createArray(int size)
    {
        srand(time(0));
        int* a = new int[size];
        int min = 0;

        for (int i = 0; i < size; i++)
        {
            a[i] = rand() % 10 + min;

            cout << a[i] << " ";

            min = a[i] + 1;
        }

        cout << endl << endl;

        return a;
    }

    void binarySearch(arr a)
    {
        int key;
        bool f = false;

        cout << "Enter key" << endl;
        cin >> key;

        int left = 0;
        int right = a.size - 1;

        while (left <= right)
        {
            int mid = left + (right - left) / 2;

            if (a.a[mid] == key)
            {
                cout << mid + 1;
                f = true;
                break;
            }
            else if (a.a[mid] < key)
            {
                left = mid + 1;
            }
            else
            {
                right = mid - 1;
            }
        }

        if (f == false)
        {
            cout << "Element not found";
        }

        cout << endl << "!!!!!! !!!!!!! !!!!!!! !!!!!!! !!!!!!!" << endl;

        binarySearch(a);
    }

    int main()

```

```

{
    int n = 0, max = -1, min = -1, key;
    arr a;

    while (n <= 0)
    {
        cout << "Enter number of elements" << endl;
        cin >> n;
    }

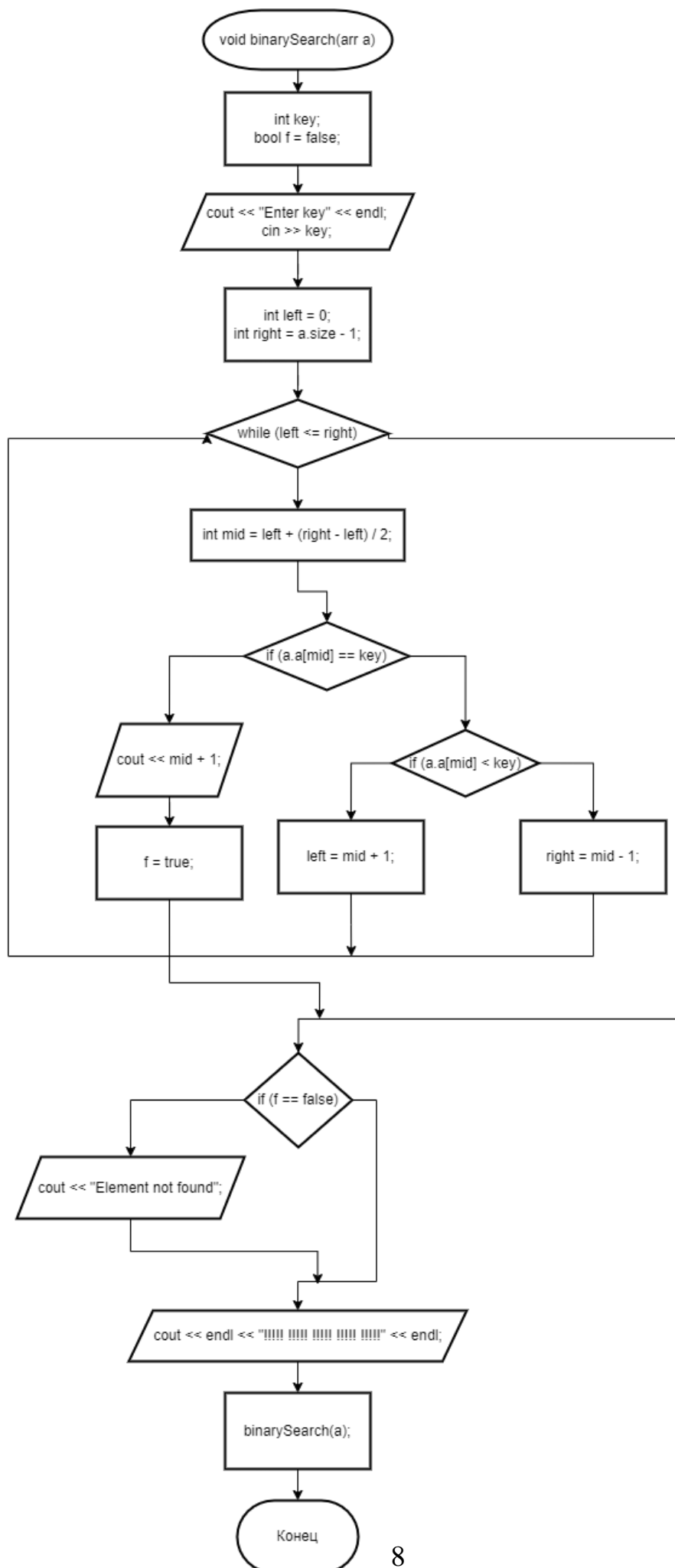
    a.a = createArray(n);
    a.size = n;

    binarySearch(a);

    return 0;
}

```

Блок-схема



Интерполяционный поиск

Код на C++

```
#include <iostream>

using namespace std;

struct arr
{
    int* a;
    int size;
};

int* createArray(int size)
{
    srand(time(0));
    int* a = new int[size];
    int min = 0;

    for (int i = 0; i < size; i++)
    {
        a[i] = rand() % 10 + min;

        cout << a[i] << " ";

        min = a[i] + 1;
    }

    cout << endl << endl;

    return a;
}

void interpolationSearch(arr a)
{
    int key;
    bool f = false;

    cout << "Enter key" << endl;
    cin >> key;

    int mid;
    int low = 0;
    int high = a.size - 1;

    while (a.a[low] < key && a.a[high] > key)
    {
        if (a.a[high] == a.a[low])
        {
            break;
        }

        mid = low + ((key - a.a[low]) * (high - low)) / (a.a[high] - a.a[low]);

        if (a.a[mid] < key)
        {
            low = mid + 1;
        }
        else if (a.a[mid] > key)
        {
            high = mid - 1;
        }
    }
}
```

```

        else
        {
            cout << mid + 1;

            f = true;

            break;
        }
    }

    if (a.a[low] == key)
    {
        cout << low + 1;

        f = true;
    }
    else if (a.a[high] == key)
    {
        cout << high + 1;

        f = true;
    }

    if (f == false)
    {
        cout << "Element not found";
    }

    cout << endl << "!!!!!! !!!!! !!!!! !!!!! !!!!!" << endl;

    interpolationSearch(a);
}

int main()
{
    int n = 0, max = -1, min = -1, key;
    arr a;

    while (n <= 0)
    {
        cout << "Enter number of elements" << endl;
        cin >> n;
    }

    a.a = createArray(n);
    a.size = n;
    interpolationSearch(a);

    return 0;
}

```

Блок-схема

