Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

направление подготовки: 09.03.01 - «Информатика и вычислительная техника»

**Отчет по Лабораторной работе «Интерполяционный поиск»**

**по дисциплине**

**«Основы алгоритмизации и программирования»**

Выполнил студент гр. ИВТ-21-1б

Ахунов Руслан Булатович

Проверил:

Ст. Преподаватель кафедры ИТАС

Яруллин Денис Владимирович

(оценка) (подпись)

г. Пермь - 2022

**Постановка задачи:**

Дан массив целых чисел (или символов, если это дано по заданию), выполнить задание из своего варианта, используя указанный метод поиска. Массив можно заполнить вручную или случайными числами, размерность массива любая (не меньше 5). Массив для интерполяционного поиска предварительно отсортировать.

Найти элемент N, удалить элемент K, стоящий перед ним.

**Анализ задачи:**

void Sort(int\* a, int n)

{

    int d;

    int tmp;

    d = n / 2;

    while (d > 0)

    {

        for (int i = 0; i < n - d; i++)

        {

            int j = i;

            while (j >= 0 && a[j] > a[j + d])

            {

                tmp = a[j];

                a[j] = a[j + d];

                a[j + d] = tmp;

                j--;

            }

        }

        d = d / 2;

    }

}

int InterSearch(int\* a, int n, int k)

{

    int left = 0;

    int right = n;

    int mid = 0;

    bool f = false;

    while ((left <= right) && !f)

    {

        mid = left + ((k - a[left]) \* (right - left)) / (a[right] - a[left]);

        if (a[mid] < k)

        {

            left = mid + 1;

        }

        else if (a[mid] > k)

        {

            right = mid + 1;

        }

        else

        {

            f = true;

        }

    }

    if (a[left] == k)

    {

        return left;

    }

    else if (a[right] == k)

    {

        return right;

    }

    return -1;

}

Так как для использования Интерполяционного поиска нужно сперва отсортировать массив, поэтому я использовал сортировку Шелла.

А сам цикл просто вычисляет по формуле область массива, где может находиться искомое. Если вычисленное не равно искомому, значит нужно сдвинуть границы области, где проходит вычисление.

Если вычисленное больше – сдвигается правая граница области поиска, если меньше – левая. Так отрезая кусок массива за куском постепенно достигается нужная ячейка массива, ну или границы области поиска сужаются до таких величин, в пределах которого уже искать нечего, когда дистанция между границами равна 1 (т.е. между точкой А и В нет более элементов для вычисления) решение говорит о том, что значение в массиве не найдено.

for (int i = ikey - 1; i < n - 1; i++)

            {

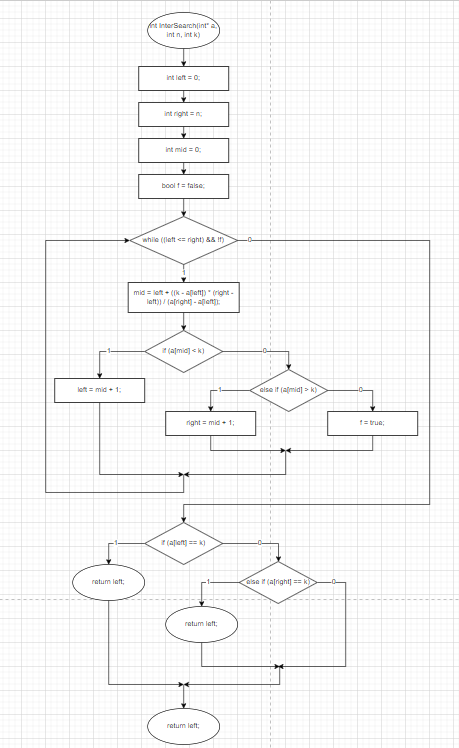
                a[i] = a[i + 1];

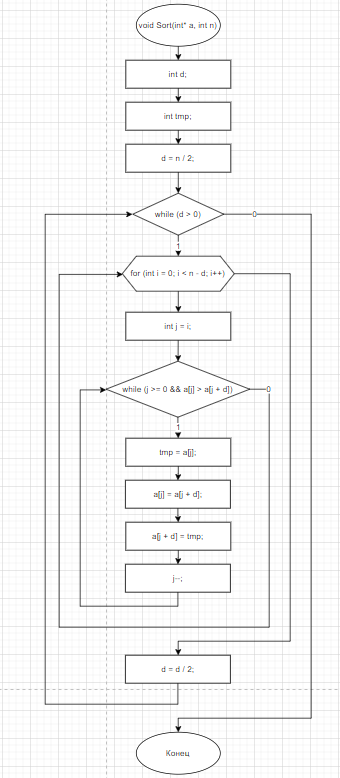
            }

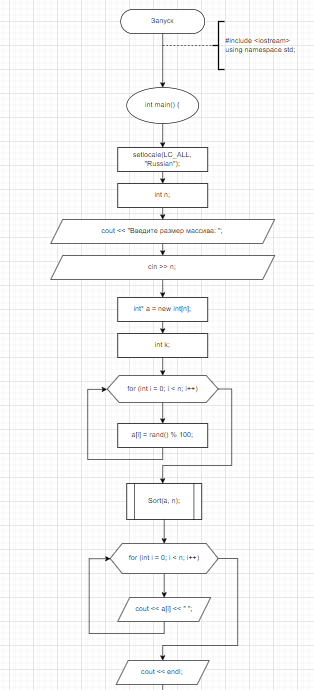
            n--;

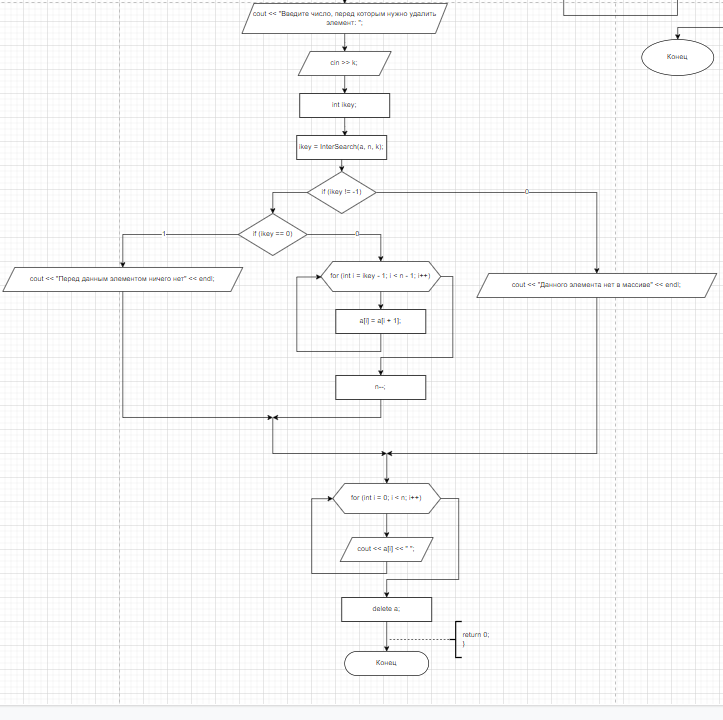
Удаляю элемент с помощью данного цикла, просто замещая его последующими значениями.

**Блок-схема**

****

****

****

****

**Код программы**

#include <iostream>

using namespace std;

void Sort(int\* a, int n)

{

    int d;

    int tmp;

    d = n / 2;

    while (d > 0)

    {

        for (int i = 0; i < n - d; i++)

        {

            int j = i;

            while (j >= 0 && a[j] > a[j + d])

            {

                tmp = a[j];

                a[j] = a[j + d];

                a[j + d] = tmp;

                j--;

            }

        }

        d = d / 2;

    }

}

int InterSearch(int\* a, int n, int k)

{

    int left = 0;

    int right = n;

    int mid = 0;

    bool f = false;

    while ((left <= right) && !f)

    {

        mid = left + ((k - a[left]) \* (right - left)) / (a[right] - a[left]);

        if (a[mid] < k)

        {

            left = mid + 1;

        }

        else if (a[mid] > k)

        {

            right = mid + 1;

        }

        else

        {

            f = true;

        }

    }

    if (a[left] == k)

    {

        return left;

    }

    else if (a[right] == k)

    {

        return right;

    }

    return -1;

}

int main()

{

    setlocale(LC\_ALL, "Russian");

    int n;

    cout << "Введите размер массива: ";

    cin >> n;

    int\* a = new int[n];

    int k;

    for (int i = 0; i < n; i++)

    {

        a[i] = rand() % 100;

    }

    Sort(a, n);

    for (int i = 0; i < n; i++)

    {

        cout << a[i] << " ";

    }

    cout << endl;

    cout << "Введите число, перед которым нужно удалить элемент: ";

    cin >> k;

    int ikey;

    ikey = InterSearch(a, n, k);

    if (ikey != -1)

    {

        if (ikey == 0)

        {

            cout << "Перед данным элементом ничего нет" << endl;

        }

        else

        {

            for (int i = ikey - 1; i < n - 1; i++)

            {

                a[i] = a[i + 1];

            }

            n--;

        }

    }

    else

    {

        cout << "Данного элемента нет в массиве" << endl;

    }

    for (int i = 0; i < n; i++)

    {

        cout << a[i] << " ";

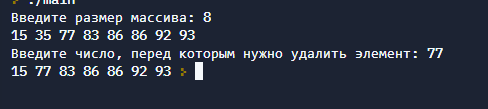
    }

    delete a;

    return 0;

}

**Вывод программы**

****