Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«**Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

Дисциплина: «Основы алгоритмизации и программирования»

Тема: Интерполяционный поиск

Выполнил работу  
Студент группы РИС-22-1Б  
Ишемцева Мария Андреевна  
Проверил  
Доцент кафедры ИТАС  
Полякова Ольга Андреевна

**Постановка задачи**

Найти в введенном массиве число интерполяционным поиском.

**Анализ переменных**

int n; - колво элементов в массиве

int a, maxi = 0; - максимальный элемент

int\* arr = new int[n]; - массив

int\* sort\_arr = new int[maxi]; - промежуточный отсортированный массив

int y = 0; - для позиции внутри массива

int find = 0; - элемент, который нужно найти

bool flag = false; - флаг отслеживающий нахождение

int left = 0, right = n-1; - границы

int fin = left + ((find - arr[left]) \* (right - left) / (arr[right] - arr[left])); - для хранения элемента, который нужно сравнить

**Анализ задачи**

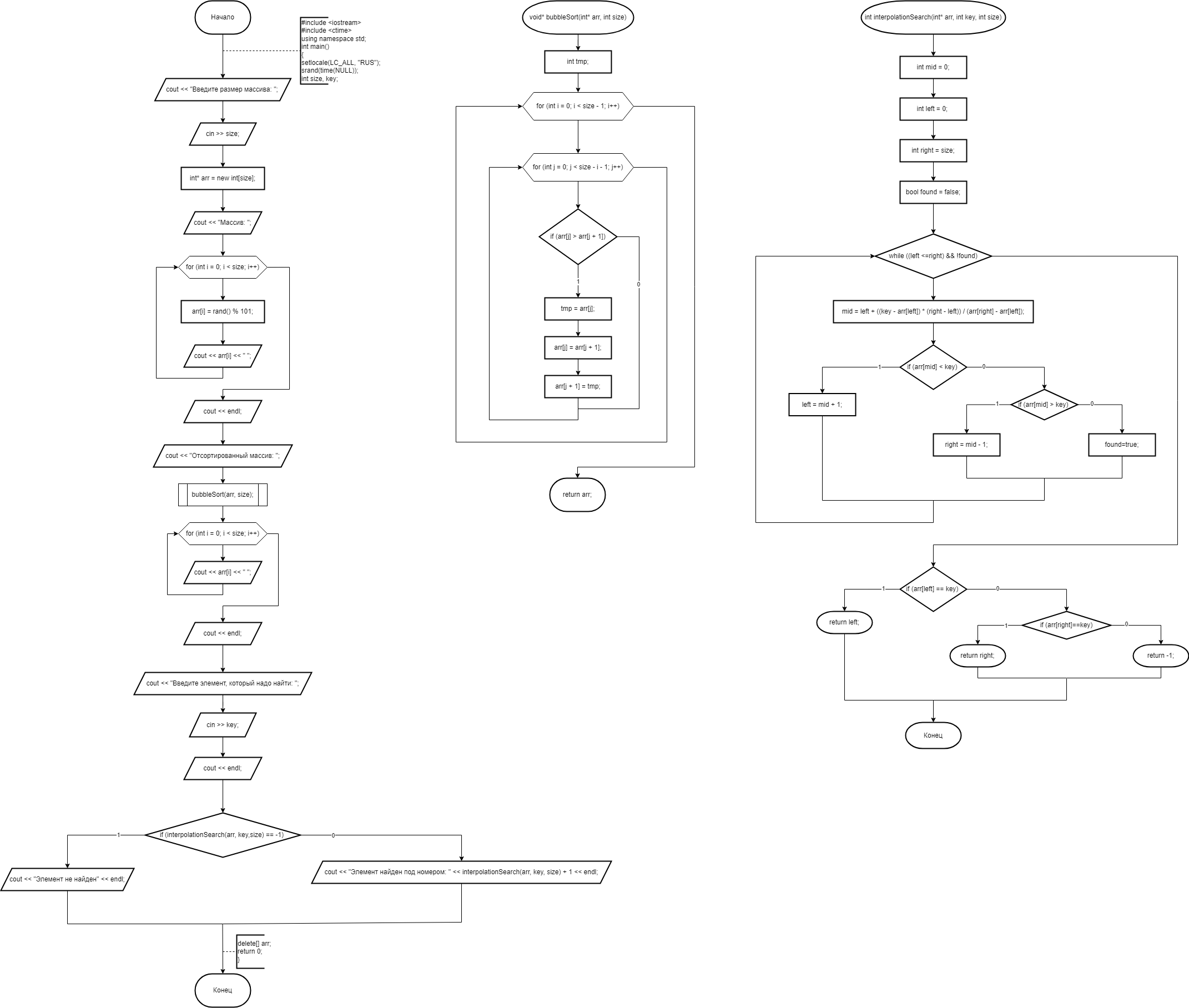
Вычисляется длина области поиска, и длина от начала области до некоего числа (скажем до центрального элемента в массиве). Вычисление это проводится как с номерами элемента, так и с их значениями, после чего полученная длина области умножается на длину между значениями, и результат прибавленный к значению из начала области дает искомое.

Формула достаточно проста – вычисляется длина между номерами первого элемента и искомого (задаваемого точнее). Такая же длина считается между первым и последним номерами. Длины между собой делятся, как раз и получая вычисление подобия. То же самое происходит со значениями элементов – так же вычисляется расстояние между граничными значениями в массиве. Специально выделяю цветом понятия и связанные с ними части формулы.

Полученная длина номеров элементов массива умножается на длину значений в этих (граничащих) элементах и прибавляется значение в первой ячейке массива.

Получается: 1 + (20-1)/(100-1) \* (200-5) = 38 с копейками.

**Блок-схема**



**Программа**

#include <iostream>

#include <string>

#include <stdio.h>

#include <stdarg.h>

using namespace std;

void interpol() {

int n;

int a, maxi = 0;

cout << "Введите количество элементов в массиве: ";

cin >> n;

int\* arr = new int[n];

cout << "Введите элементы массива" << endl;

for (int i = 0; i < n; i++) {

cin >> a;

if (a > maxi) maxi = a;

arr[i] = a;

}

int\* sort\_arr = new int[maxi];

for (int i = 1; i < maxi + 1; i++) {

sort\_arr[i] = 0;

}

for (int i = 0; i < n; i++) {

sort\_arr[arr[i]]++;

}

int y = 0;

for (int i = 1; i < maxi + 1; i++) {

if (sort\_arr[i] > 0) {

for (int j = 0; j < sort\_arr[i]; j++) {

arr[y] = i;

y++;

}

}

}

cout << "Введите элемент, который нужно найти: ";

int find = 0;

cin >> find;

//fin = (((find-1)\*(n - 1))/(maxi-1)) + 1

bool flag = false;

int left = 0, right = n-1;

int fin = left + ((find - arr[left]) \* (right - left) / (arr[right] - arr[left]));

cout << "Отсортированный массив: ";

for (int i = 0; i < n; i++) {

cout << arr[i] << " ";

}

cout << endl;

while (!flag && left < right - 1) {

if (arr[fin] != find) {

if (arr[fin] > find) {

right = fin;

fin = left + ((find - arr[left]) \* (right - left) / (arr[right] - arr[left]));

}

else {

left = fin+1;

fin = left + ((find - arr[left]) \* (right - left) / (arr[right] - arr[left]));

}

}

else {

flag = true;

}

}

if (flag) cout << "Индекс искомого элемента: " << fin << endl;

else cout << "Элемент не найден!" << endl;

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

interpol();

return 0;

}

**Вывод программы**

