ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа

Сортировки

Сортировка слиянием

Выполнил студент группы РИС-23-3Б

Л. А. Зарайский

Проверила доцент кафедры ИТАС

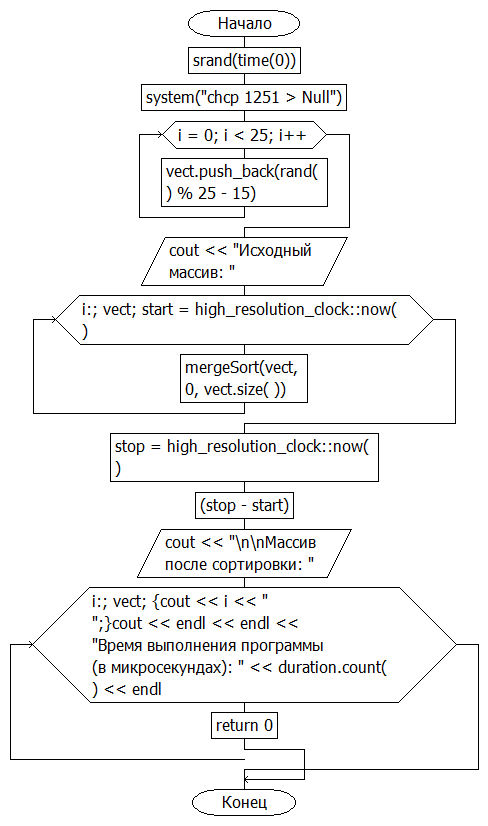
О. А. Полякова

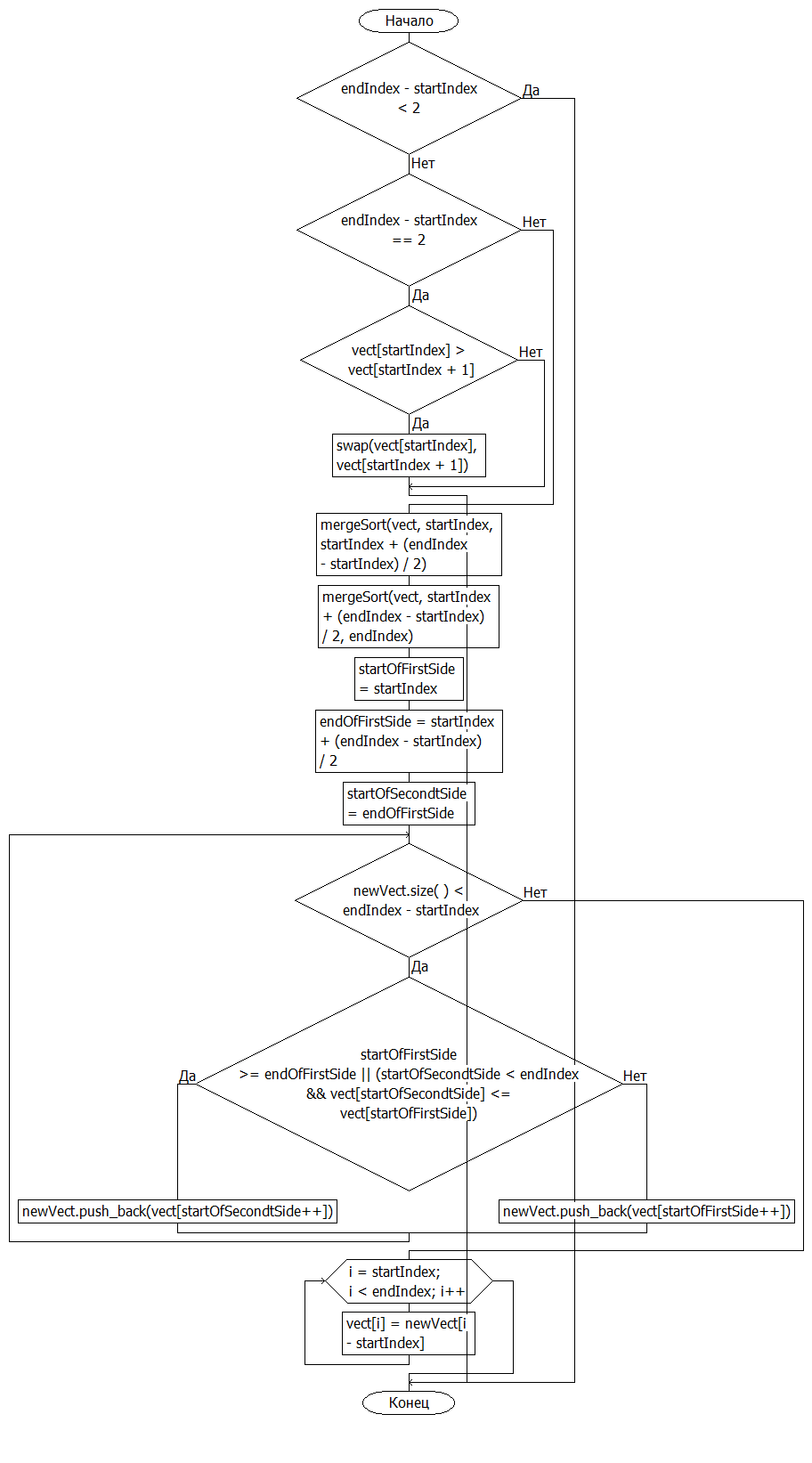
2024 г.

**Постановка задачи**

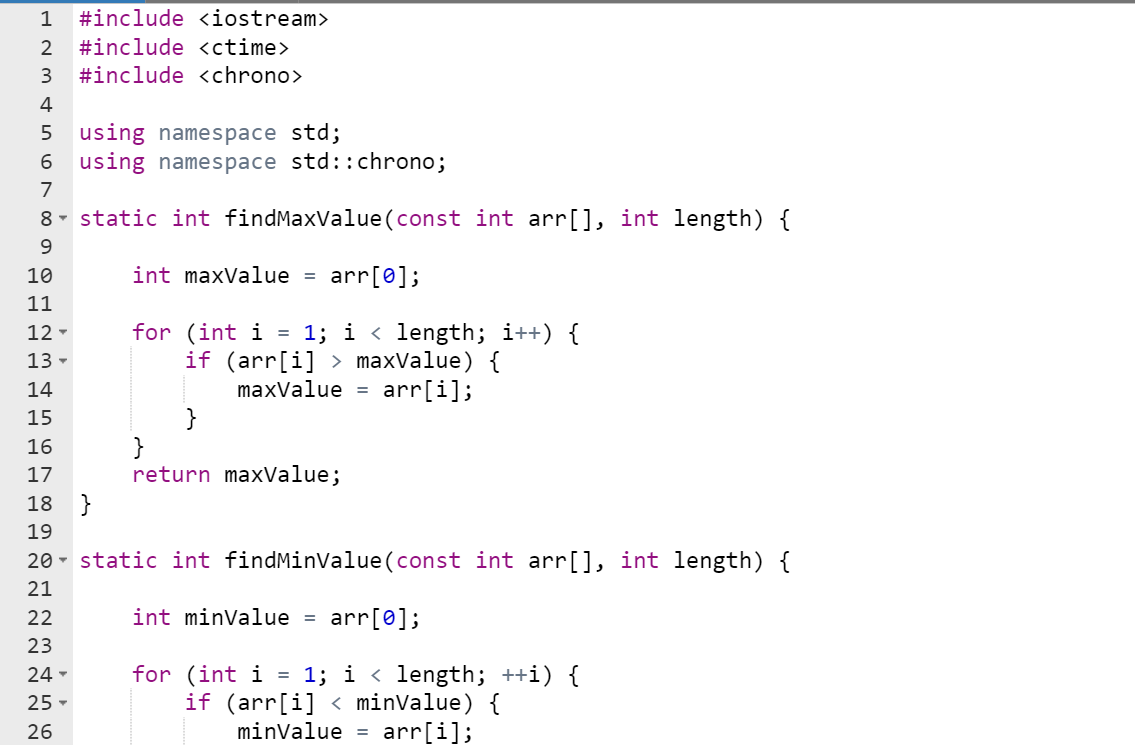
Дан массив из 25 элементов. Отсортировать его при помощи сортировки слиянием, засечь время выполнения программы.

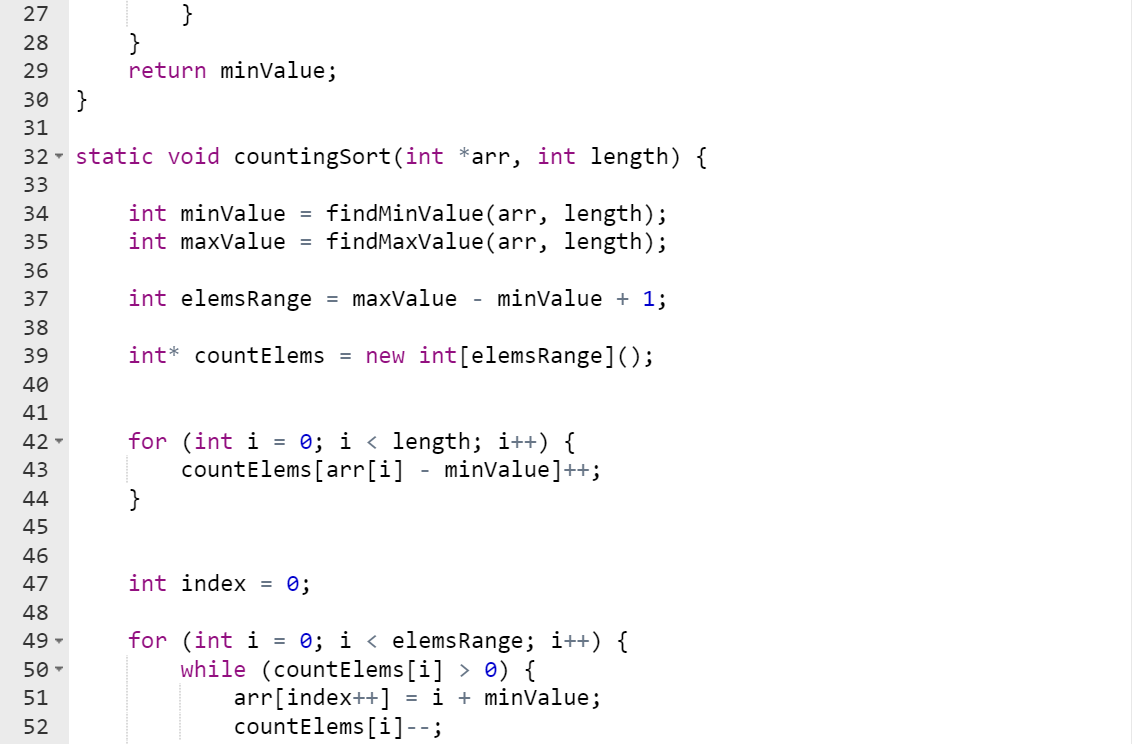
**Блок-схема**

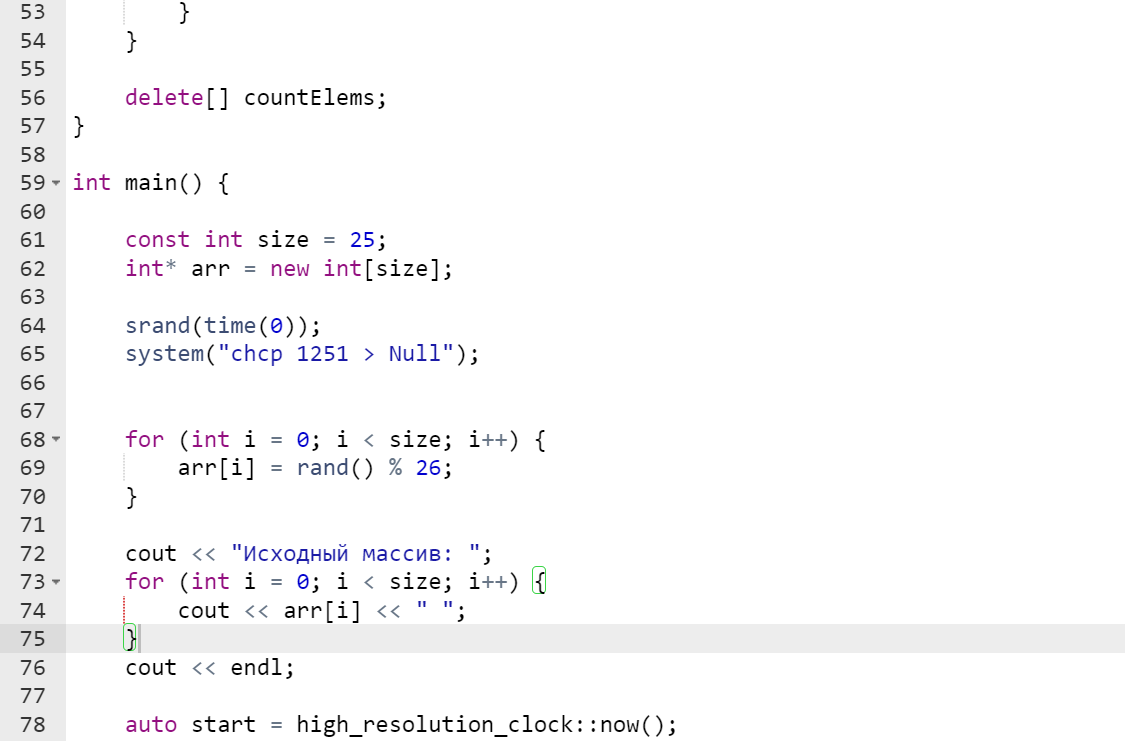
****

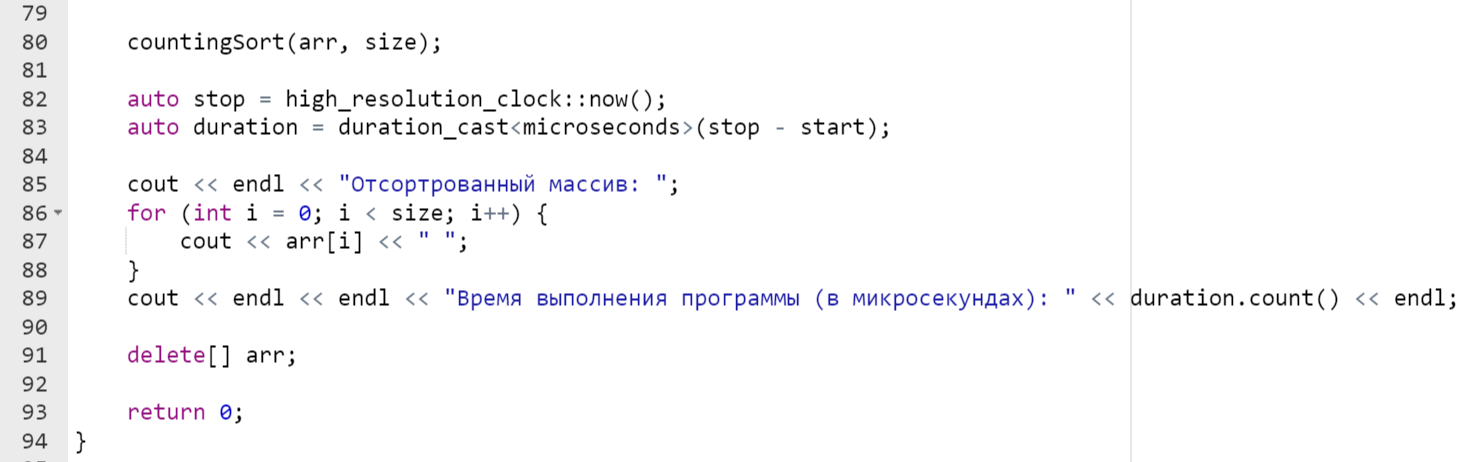
****

**Код на языке C++**

****

****

****

****

﻿#include <iostream>

#include <vector>

#include <ctime>

#include <chrono>

using namespace std;

using namespace std::chrono;

static void mergeSort(vector<int>& vect, int startIndex, int endIndex)

{

if (endIndex - startIndex < 2)

return;

if (endIndex - startIndex == 2)

{

if (vect[startIndex] > vect[startIndex + 1])

swap(vect[startIndex], vect[startIndex + 1]);

return;

}

mergeSort(vect, startIndex, startIndex + (endIndex - startIndex) / 2);

mergeSort(vect, startIndex + (endIndex - startIndex) / 2, endIndex);

vector<int> newVect;

int startOfFirstSide = startIndex;

int endOfFirstSide = startIndex + (endIndex - startIndex) / 2;

int startOfSecondtSide = endOfFirstSide;

while (newVect.size() < endIndex - startIndex)

{

if (startOfFirstSide >= endOfFirstSide ||

(startOfSecondtSide < endIndex && vect[startOfSecondtSide] <= vect[startOfFirstSide]))

{

newVect.push\_back(vect[startOfSecondtSide++]);

}

else

{

newVect.push\_back(vect[startOfFirstSide++]);

}

}

for (int i = startIndex; i < endIndex; i++)

vect[i] = newVect[i - startIndex];

}

int main()

{

srand(time(0));

system("chcp 1251 > Null");

vector<int> vect;

for (int i = 0; i < 25; i++) {

vect.push\_back(rand() % 25 - 15);

}

cout << "Исходный массив: ";

for (auto i : vect) {

cout << i << " ";

}

auto start = high\_resolution\_clock::now();

mergeSort(vect, 0, vect.size());

auto stop = high\_resolution\_clock::now();

auto duration = duration\_cast<microseconds>(stop - start);

cout << "\n\nМассив после сортировки: ";

for (auto i : vect) {

cout << i << " ";

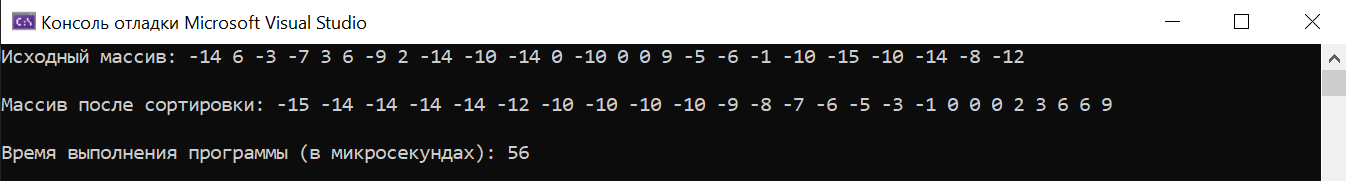
}

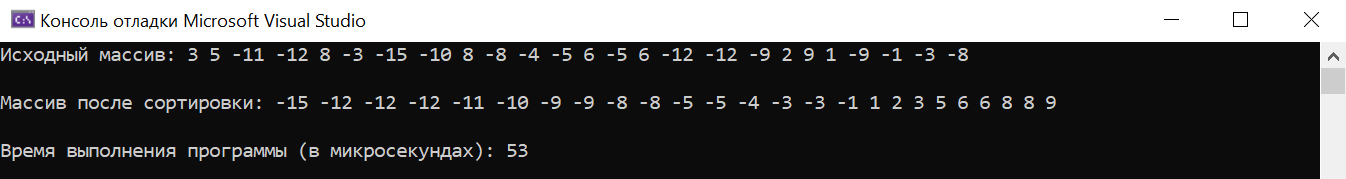
cout << endl << endl << "Время выполнения программы (в микросекундах): " << duration.count() << endl;

return 0;

}

**Результаты выполнения программы**

****

****

**Выводы**

Программа работает успешно.  
 Ссылка на GitHub: