ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа

Сортировки

Блочная сортировка

Выполнил студент группы РИС-23-3Б

Л. А. Зарайский

Проверила доцент кафедры ИТАС

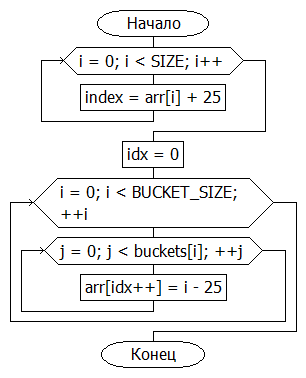
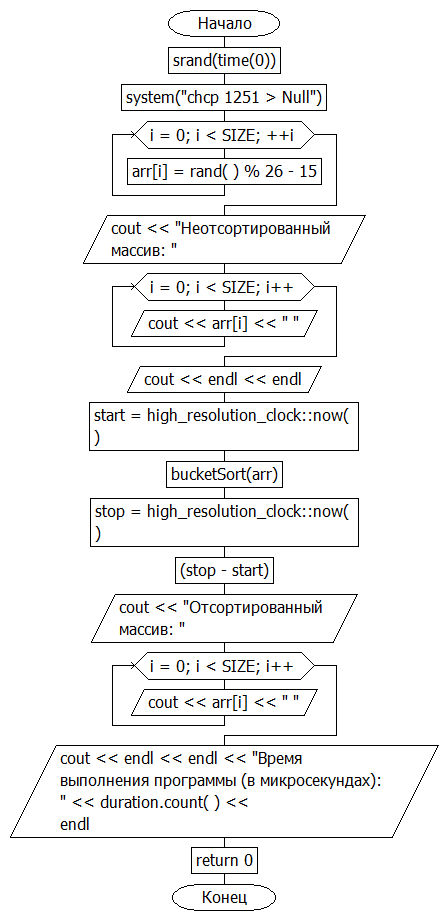
О. А. Полякова

2024 г.

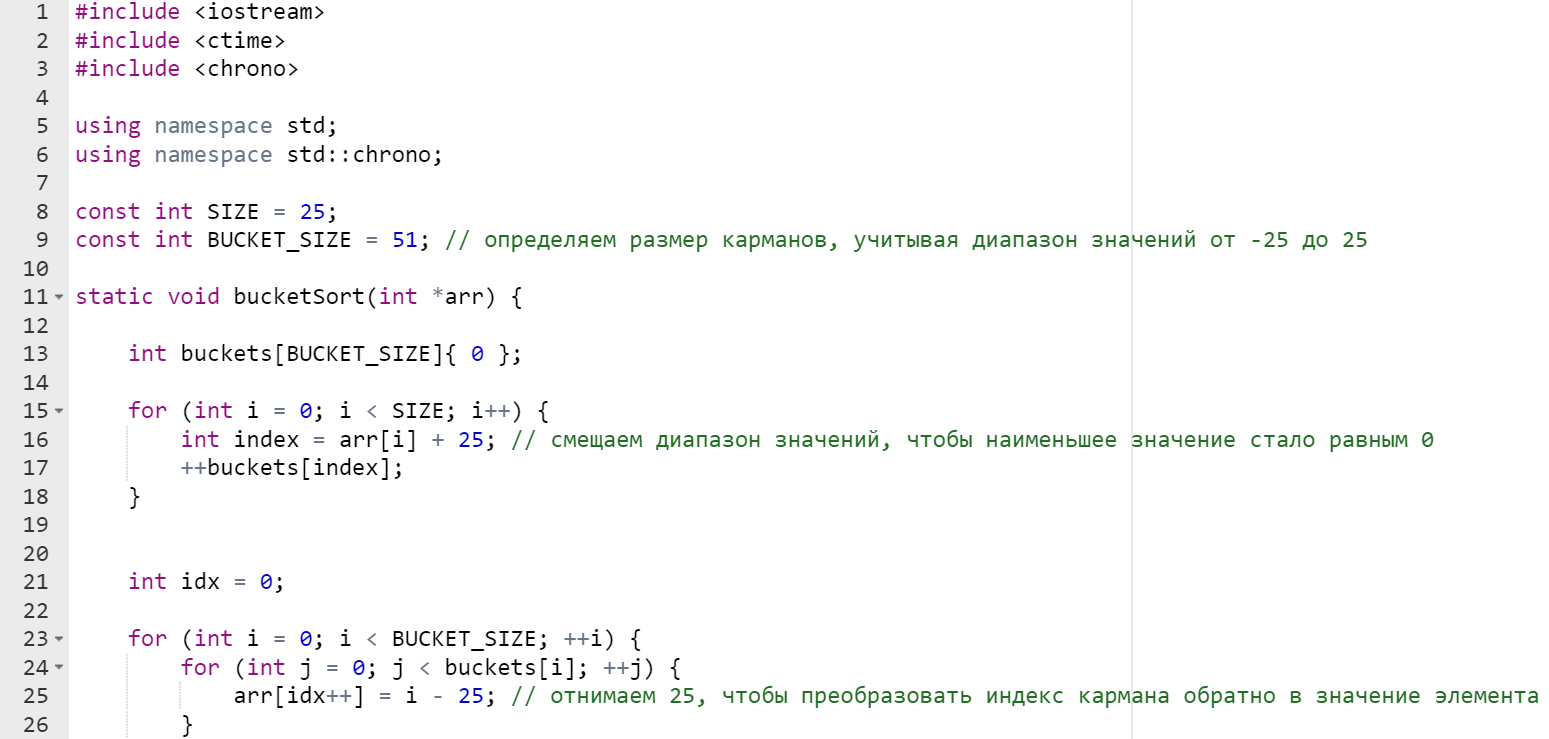
**Постановка задачи**

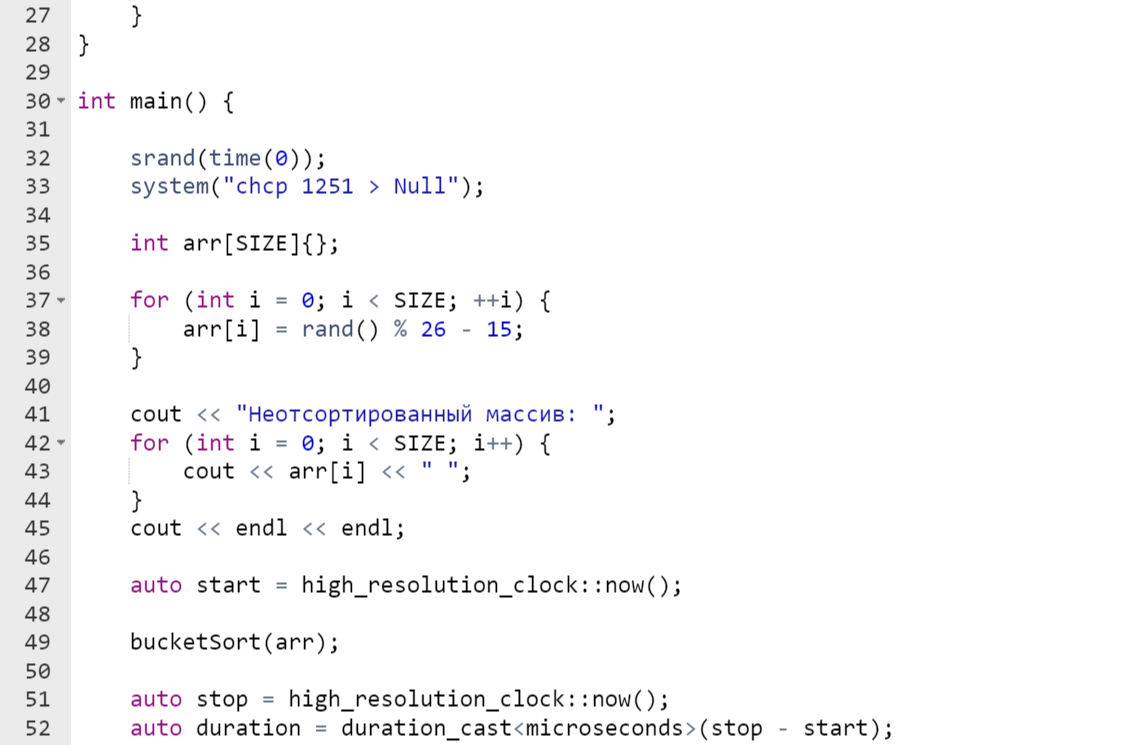
Дан массив из 25 элементов, отсортировать его с помощью блочной сортировки, измерить время выполнения программы.

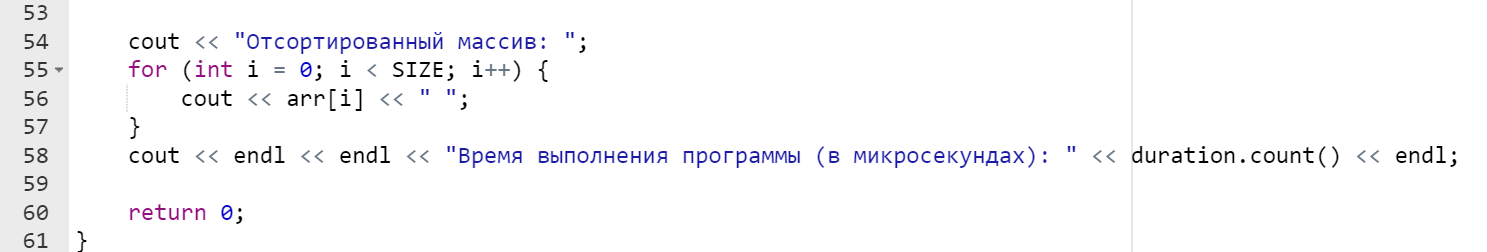
**Блок-схема**

****

**Код на языке C++**

****

****

****

﻿#include <iostream>

#include <ctime>

#include <chrono>

using namespace std;

using namespace std::chrono;

const int SIZE = 25;

const int BUCKET\_SIZE = 51; // определяем размер карманов, учитывая диапазон значений от -25 до 25

static void bucketSort(int \*arr) {

int buckets[BUCKET\_SIZE]{ 0 };

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

int index = arr[i] + 25; // смещаем диапазон значений, чтобы наименьшее значение стало равным 0

++buckets[index];

}

int idx = 0;

for (int i = 0; i < BUCKET\_SIZE; ++i) {

for (int j = 0; j < buckets[i]; ++j) {

arr[idx++] = i - 25; // отнимаем 25, чтобы преобразовать индекс кармана обратно в значение элемента

}

}

}

int main() {

srand(time(0));

system("chcp 1251 > Null");

int arr[SIZE]{};

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

arr[i] = rand() % 26 - 15;

}

cout << "Неотсортированный массив: ";

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

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

}

cout << endl << endl;

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

bucketSort(arr);

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

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

cout << "Отсортированный массив: ";

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

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

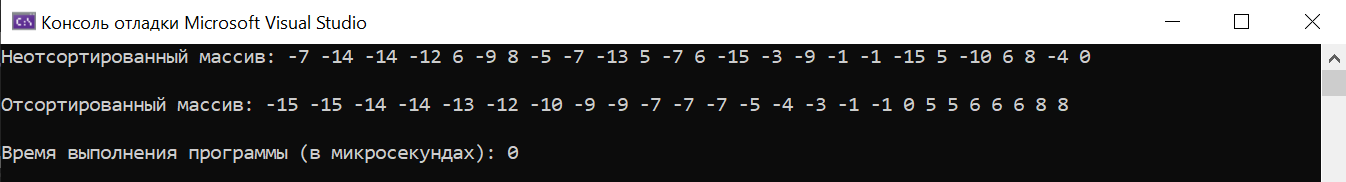
}

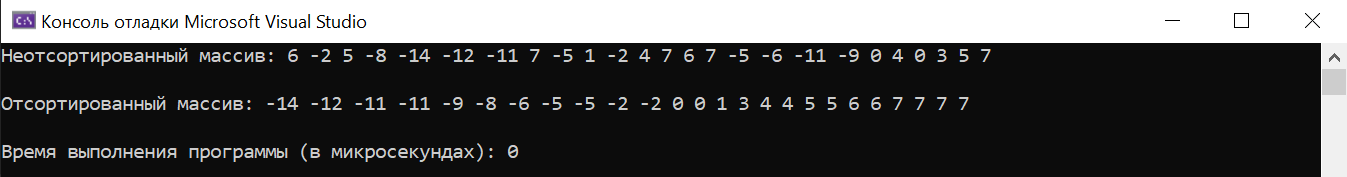
cout << endl << endl << "Время выполнения программы (в микросекундах): " << duration.count() << endl;

return 0;

}

Результаты работы программы

****

****

**Выводы**

Программа работает успешно.

Ссылка на GitHub: