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: 09.03.04 – «Программная инженерия»

**Лабораторная работа.**

**«Блочная сортировка»**

Выполнил студент гр. РИС-24-1б

Чижов Денис Николаевич

Проверил:

Доц. Каф. ИТАС 

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2025

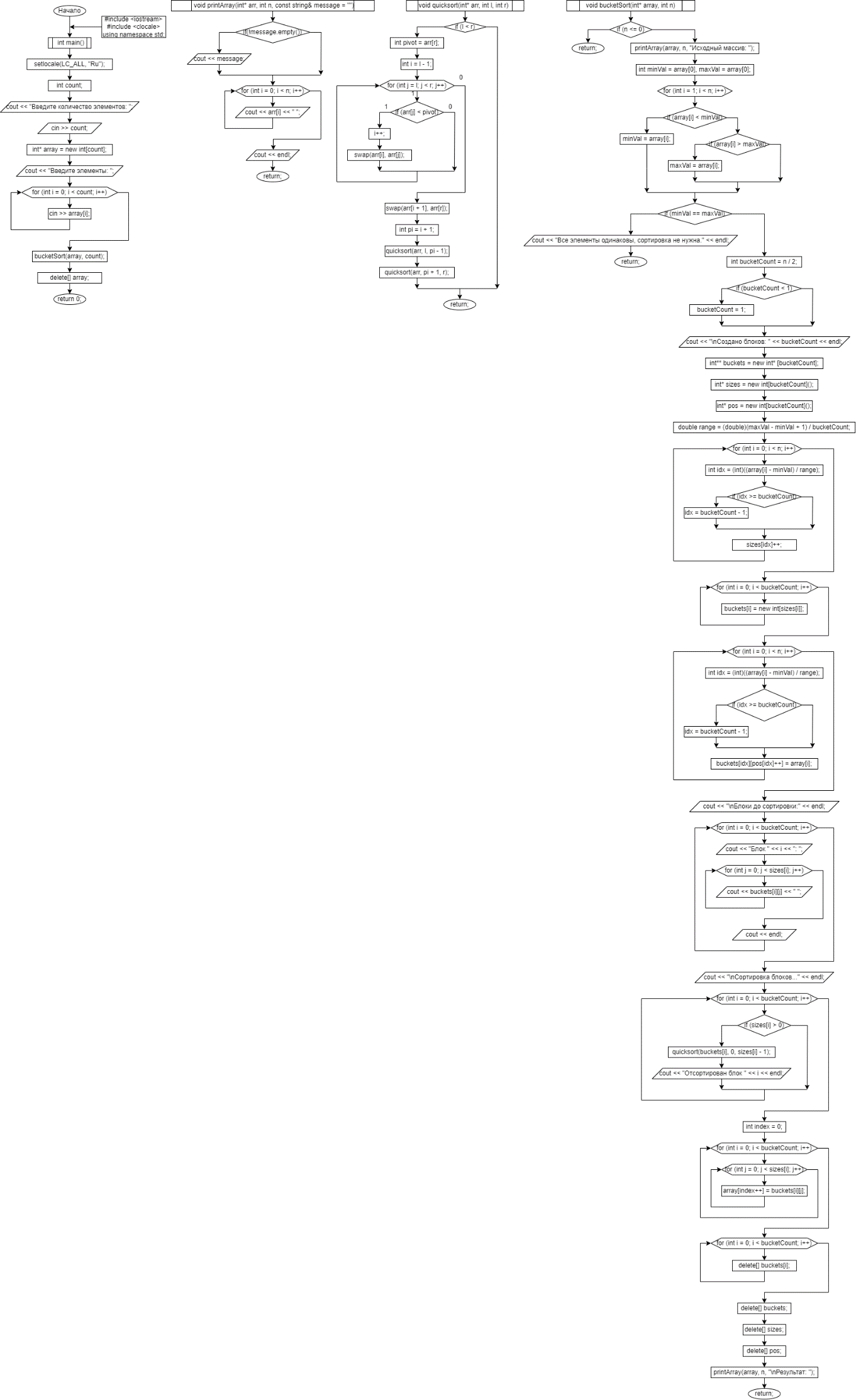
**Задача:**

Отсортировать массив блочной сортировкой.

**Анализ и визуализация:**

Сортируем массив блочной сортировкой.

**Блок-схема:**



**Код:**

#include <iostream>

#include <clocale>

using namespace std;

// Функция для вывода массива

void printArray(int\* arr, int n, const string& message = "") {

if (!message.empty()) cout << message;

for (int i = 0; i < n; i++) cout << arr[i] << " ";

cout << endl;

}

// Функция QuickSort

void quicksort(int\* arr, int l, int r) {

if (l < r) {

int pivot = arr[r];

int i = l - 1;

for (int j = l; j < r; j++) {

if (arr[j] < pivot) {

i++;

swap(arr[i], arr[j]);

}

}

swap(arr[i + 1], arr[r]);

int pi = i + 1;

quicksort(arr, l, pi - 1);

quicksort(arr, pi + 1, r);

}

}

void bucketSort(int\* array, int n) {

if (n <= 0) return;

printArray(array, n, "Исходный массив: ");

// Находим min и max

int minVal = array[0], maxVal = array[0];

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

if (array[i] < minVal) minVal = array[i];

else if (array[i] > maxVal) maxVal = array[i];

}

// Если все элементы одинаковые

if (minVal == maxVal) {

cout << "Все элементы одинаковы, сортировка не нужна." << endl;

return;

}

// Создаем блоки

int bucketCount = n / 2;

if (bucketCount < 1) bucketCount = 1;

cout << "\nСоздано блоков: " << bucketCount << endl;

// Инициализация блоков

int\*\* buckets = new int\* [bucketCount];

int\* sizes = new int[bucketCount]();

int\* pos = new int[bucketCount]();

// Распределяем элементы по блокам

double range = (double)(maxVal - minVal + 1) / bucketCount;

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

int idx = (int)((array[i] - minVal) / range);

if (idx >= bucketCount) idx = bucketCount - 1;

sizes[idx]++;

}

// Выделяем память

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

buckets[i] = new int[sizes[i]];

}

// Заполняем блоки

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

int idx = (int)((array[i] - minVal) / range);

if (idx >= bucketCount) idx = bucketCount - 1;

buckets[idx][pos[idx]++] = array[i];

}

// Выводим блоки до сортировки

cout << "\nБлоки до сортировки:" << endl;

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

cout << "Блок " << i << ": ";

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

cout << buckets[i][j] << " ";

}

cout << endl;

}

// Сортируем блоки

cout << "\nСортировка блоков..." << endl;

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

if (sizes[i] > 0) {

quicksort(buckets[i], 0, sizes[i] - 1);

cout << "Отсортирован блок " << i << endl;

}

}

// Объединяем блоки

int index = 0;

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

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

array[index++] = buckets[i][j];

}

}

// Освобождаем память

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

delete[] buckets[i];

}

delete[] buckets;

delete[] sizes;

delete[] pos;

printArray(array, n, "\nРезультат: ");

}

int main() {

setlocale(LC\_ALL, "Ru");

int count;

cout << "Введите количество элементов: ";

cin >> count;

int\* array = new int[count];

cout << "Введите элементы: ";

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

cin >> array[i];

}

bucketSort(array, count);

delete[] array;

return 0;

}

**Вывод:**

Программа работает. Задача решена.