Министерство высшего образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет» (ПНИПУ)**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

ОТЧЁТ

о лабораторной работе

Тема: «Реализация быстрых методов сортировок»

Выполнил

Студент группы РИС-22-2б

Прядеин И.А.

Проверил доцент кафедры

ИТАС

Полякова О. А.

Пермь 2023

**Постановка задачи**

Реализовать метод «блочной» сортировки, метод «слияния», «быстрый метод», метод «подсчётом» с помощью языка программирования C++.

**Исходный код программы («Быстрый» метод):**

#include <iostream>

#include <ctime>

using namespace std;

void ShowArray(int\* Arr, int SIZE)

{

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

cout << Arr[i] << " ";

cout << endl;

}

void quickSort(int\* array, int low, int high)

{

int i = low;

int j = high;

int pivot = array[(i + j) / 2];

int temp;

while (i <= j)

{

while (array[i] < pivot)

i++;

while (array[j] > pivot)

j--;

if (i <= j)

{

temp = array[i];

array[i] = array[j];

array[j] = temp;

i++;

j--;

}

}

if (j > low)

quickSort(array, low, j);

if (i < high)

quickSort(array, i, high);

}

int main()

{

setlocale(LC\_ALL, "Russian");

srand(time(NULL));

int SIZE = 10;

int\* Array = new int[SIZE];

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

Array[i] = rand() % 10;

cout << "Массив до сотрировки: ";

ShowArray(Array, SIZE);

cout << "Метод быстрой сортировки: ";

quickSort(Array, 0, SIZE - 1);

ShowArray(Array, SIZE);

delete[] Array;

return 0;

}

**Исходный код программы («Блочный» метод):**

#include <iostream>

#include <algorithm>

#include <vector>

#include <ctime>

using namespace std;

void ShowArray(float Arr[], int SIZE)

{

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

cout << Arr[i] << " ";

cout << endl;

}

void BucketSort(float Arr[], int size)

{

vector<float> buckets[size];

for (int i = 0; i < size; i++)

{

int bucketIndex = size \* Arr[i];

buckets[bucketIndex].push\_back(Arr[i]);

}

for (int i = 0; i < size; i++)

sort(buckets[i].begin(), buckets[i].end());

int index = 0;

for (int i = 0; i < size; i++)

for (int j = 0; j < buckets[i].size(); j++, index++)

Arr[index] = buckets[i][j];

}

int main()

{

setlocale(LC\_ALL, "Russian");

srand(time(NULL));

const int SIZE = 10;

float Array[SIZE] = { 0.23, 0.53, 0.13, 0.6, 0.43, 0.32, 0.54, 0.96, 0.65, 0.21 };

cout << "Массив до сотрировки: ";

ShowArray(Array, SIZE);

cout << "Блочный метод сортировки: ";

BucketSort(Array, SIZE);

ShowArray(Array, SIZE);

return 0;

}

**Исходный код программы (Метод слияния):**

#include <iostream>

#include <ctime>

using namespace std;

void ShowArray(int\* Arr, int SIZE)

{

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

cout << Arr[i] << " ";

cout << endl;

}

void Merge(int\* Arr, int first, int last)

{

int middle, start, final, j;

int\* mas = new int[100];

middle = (first + last) / 2;

start = first;

final = middle + 1;

for (j = first; j <= last; j++)

if ((start <= middle) && ((final > last) || (Arr[start] < Arr[final])))

{

mas[j] = Arr[start];

start++;

}

else

{

mas[j] = Arr[final];

final++;

}

for (j = first; j <= last; j++)

Arr[j] = mas[j];

delete[] mas;

}

void MergeSort(int\* Arr, int first, int last)

{

if (first < last)

{

MergeSort(Arr, first, (first + last) / 2);

MergeSort(Arr, (first + last) / 2 + 1, last);

Merge(Arr, first, last);

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

srand(time(NULL));

int SIZE = 10;

int\* Array = new int[SIZE];

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

Array[i] = rand() % 10;

cout << "Массив до сотрировки: ";

ShowArray(Array, SIZE);

cout << "Метод сортировки слиянием: ";

MergeSort(Array, 0, SIZE - 1);

ShowArray(Array, SIZE);

delete[] Array;

return 0;

}

**Исходный код программы (Сортировка «подсчётом»):**

#include <iostream>

#include <ctime>

using namespace std;

void ShowArray(int\* Arr, int SIZE)

{

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

cout << Arr[i] << " ";

cout << endl;

}

void CountSort(int\* Array, int size, int MaxRange)

{

int TempArr[MaxRange + 1];

for (int i = 0; i < MaxRange + 1; i++)

TempArr[i] = 0;

for (int i = 0; i < size; i++)

TempArr[Array[i]]++;

int index = 0;

for (int i = 0; i < MaxRange + 1; i++)

while (TempArr[i]--)

Array[index++] = i;

}

int main()

{

setlocale(LC\_ALL, "Russian");

srand(time(NULL));

int SIZE = 10;

int\* Array = new int[SIZE];

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

Array[i] = rand() % 11;

cout << "Массив до сотрировки: ";

ShowArray(Array, SIZE);

cout << "Метод сортировки подсчётом: ";

CountSort(Array, SIZE, 10);

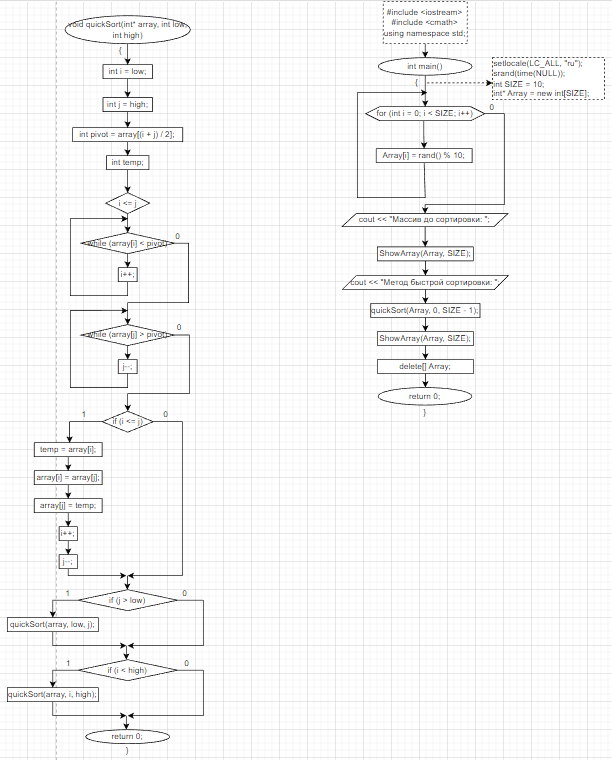
ShowArray(Array, SIZE);

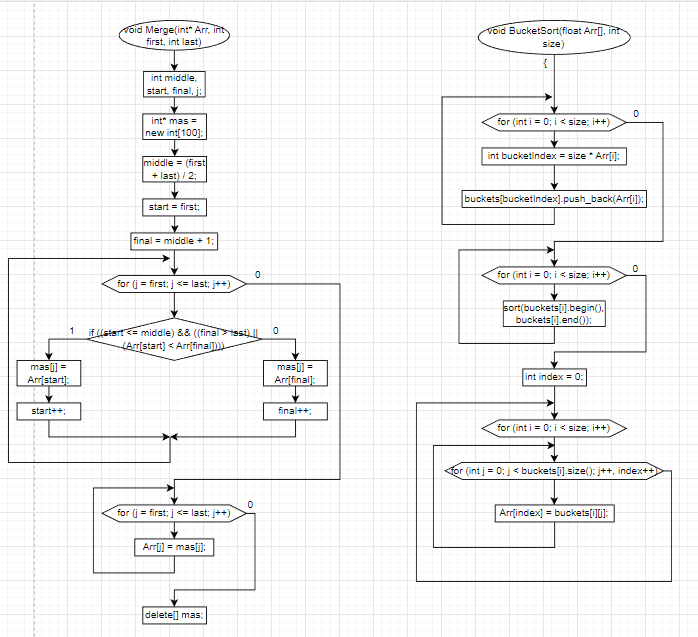
delete[] Array;

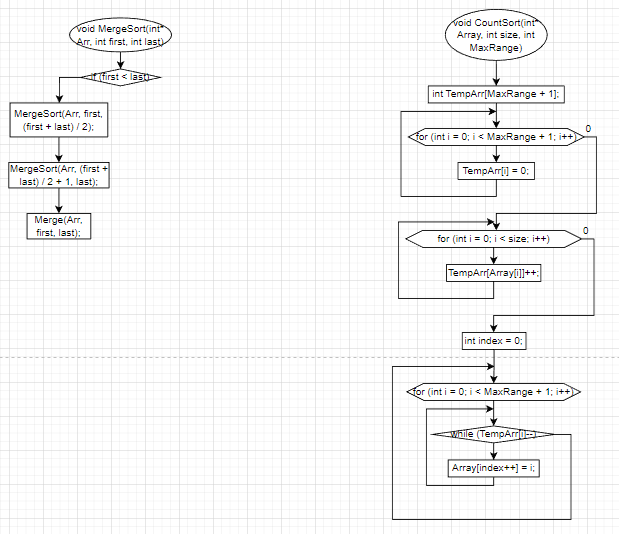
return 0;

}

**Блок схема**







**Скриншоты результатов выполнения программы**









