Федеральное государственное автономное образовательное учреждение

высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

КАФЕДРА СИСТЕМНОГО АНАЛИЗА И ТЕЛЕКОММУНИКАЦИИ

**ПРОВЕРИЛ ВЫПОЛНИЛ**

Ассистент кафедры САиТ Студент группы КТбо1-2

Лапшин В.С. Макаричев СВ

«21» февраля\_2023 г. «21» \_февраля\_\_2023 г.

**ЛАБОРАТОРНАЯ РАБОТА №1**

По дисциплине

«Основы алгоритмизации и программирования»

Вариант 4

На тему

«Сортировки»

**Таганрог 2023**

**Цель**: Написать программу, реализующую несколько алгоритмов сортировок, и провести их сравнительный анализ на одном входном наборе данных.

**Задачи**: Для выполнения лабораторной работы необходимо подготовить исходный файл данных следующим образом. Написать программу, которая должна сгенерировать массив и з 1000000 (один миллион) случайных целых чисел и записать их в файл input.txt. Данный файл представляет собой исходный массив для сортировки.

**Ход работы**:

**Составление математической модели**: *(Пишем формулу)*

** *(Блок-схема программы.)*

**Реализация .**

#include <iostream>

#include <algorithm>

#include <fstream>

#include <vector>

#include <string>

#include <chrono>

const int N = 10000;

using namespace std::chrono;

std::vector <int> array(N);

std::vector <int> additional(N);

int CounterForQuick = 0;

void FillArr();

void BubbleSort();

void QuickSort(int size);

void GnomSort();

int main()

{

std::ofstream fout("result.txt");

FillArr();

std::copy(array.begin(), array.end(), additional.begin());

auto begin = steady\_clock::now();

GnomSort();

auto end = steady\_clock::now();

std::cout << "Time for BubleSort - "<< duration\_cast<nanoseconds>(end - begin).count() << std::endl;

std::copy(additional.begin(), additional.end(), array.begin());

begin = steady\_clock::now();

BubbleSort();

end = steady\_clock::now();

std::cout << "Time for GnomSort - " << duration\_cast<nanoseconds>(end - begin).count() << std::endl;

std::copy(additional.begin(), additional.end(), array.begin());

begin = steady\_clock::now();

QuickSort(array.size());

end = steady\_clock::now();

std::cout << "Count for Quick Movement - " << CounterForQuick << std::endl;

std::cout << "Time for QuickSort - " << duration\_cast<nanoseconds>(end - begin).count() << std::endl;

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

{

fout << array[i] << std::endl;

}

}

void FillArr()

{

std::ifstream fin("for\_sort.txt");

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

{

fin >> array[i];

}

}

void BubbleSort()

{

int CounterForBubble = 0;

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

{

for (int j = 0; j < N-1; j++)

{

if (array[j] > array[j + 1])

{

std::swap(array[j], array[j + 1]);

CounterForBubble++;

}

}

}

std::cout << "Count for Bubble Movement - " << CounterForBubble << std::endl;

}

void QuickSort(int size)

{

//Указатели в начало и в конец массива

int i = 0;

int j = size - 1;

//Центральный элемент массива

int mid = array[size / 2];

//Делим массив

do {

//Пробегаем элементы, ищем те, которые нужно перекинуть в другую часть

//В левой части массива пропускаем(оставляем на месте) элементы, которые меньше центрального

while (array[i] < mid) {

i++;

}

//В правой части пропускаем элементы, которые больше центрального

while (array[j] > mid) {

j--;

}

//Меняем элементы местами

if (i <= j) {

int tmp = array[i];

array[i] = array[j];

array[j] = tmp;

CounterForQuick++;

i++;

j--;

}

} while (i <= j);

//Рекурсивные вызовы, если осталось, что сортировать

if (j > 0) {

//"Левый кусок"

QuickSort( j + 1);

}

if (i < size) {

//"Првый кусок"

QuickSort(size - i);

}

}

void GnomSort()

{

int CounterForGnom = 0;

for (int i = 0; i < N-1;)

{

if (array[i + 1] < array[i] )

{

std::swap(array[i + 1], array[i]);

CounterForGnom++;

if (i != 0)

{

i--;

}

}

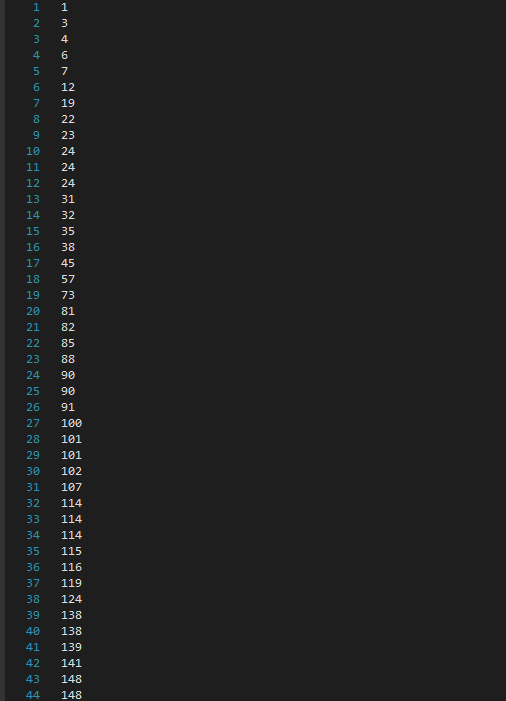
else { i++; }

}

std::cout << "Count for Gnom Movement - " << CounterForGnom << std::endl;

}

**Результат работы программы.**

**

**Вывод:** Рабочий код программы оценивающий 3 вида сортировки на одних и тех же вводных данных, оценка алгоритмов производится по количеству операций и времени.