Федеральное государственное образовательное учреждение

высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

КАФЕДРА СИСТЕМНОГО АНАЛИЗА И ИНТЕГРАЦИИ IT-РЕШЕНИЙ

**ПРОВЕРИЛ ВЫПОЛНИЛ**

Доцент кафедры САИТ Студент группы КТбо1-2

Лапшин Вячеслав Сергеев Хиль Дмитрий Сергеевич

«25» февраля 2023 г. «24» февраля 2023 г.

**ЛАБОРАТОРНАЯ РАБОТА №1**

По дисциплине

«Основы алгоритмизации и

программирования»

Вариант 5

На тему

«Алгоритмы сортировки»

**Цель**: Цель данной лабораторной работы - ознакомиться с сортировкой данных и ее видами, а также запрограммировать на языке Си некоторые из алгоритмов сортировок в соответствии с индивидуальным заданием.

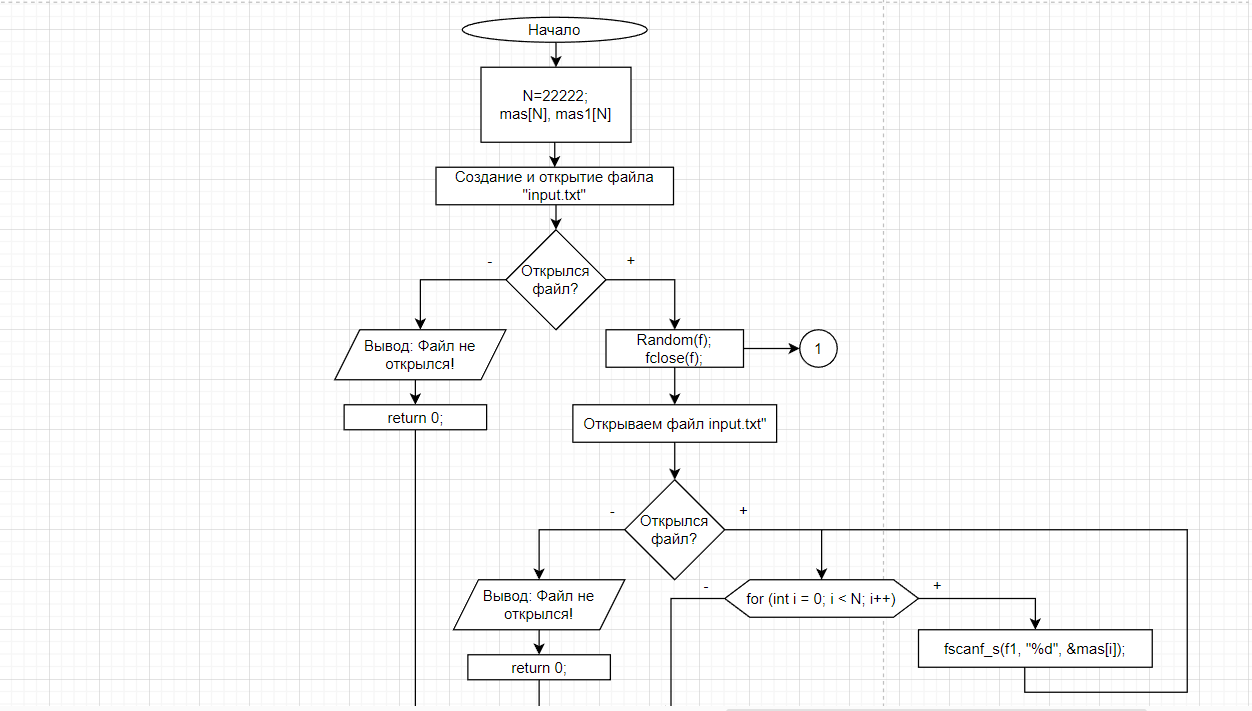
**Задачи**: написать программу, реализующую несколько алгоритмов сортировок, и провести их сравнительный анализ на одном входном наборе данных. Для выполнения лабораторной работы необходимо подготовить исходный файл данных следующим образом. 10 Написать программу, которая должна сгенерировать массив и з 1000000 (один миллион) случайных целых чисел и записать их в файл input.txt. Данный файл представляет собой исходный массив для сортировки.

5. Произвести сортировку алгоритмами: сортировка пузырьком, сортировка слиянием, сортировка Шелла.

**Составление математической модели:**

Создаю операции. Первая операция - запись в файл input.txt рандомных значений. Вторая операция - сортировка пузырьком. Третья операция - сортировка Шелла. Четвертая операция - сортировка слиянием. В main происходит извлечение данных файла в массив. Происходит замер времени работы каждой сортировки. В конце делаем запись отсортированного массива в файл output.txt.

**Ход работы**:



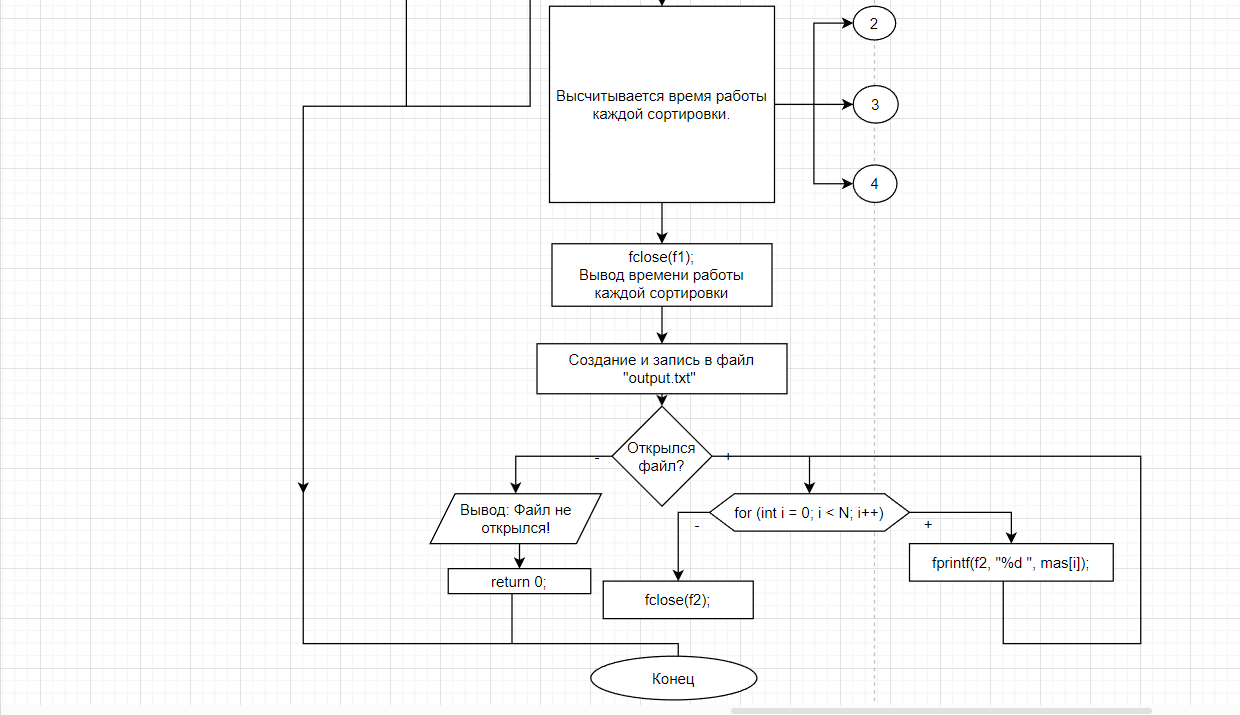


Рисунок 1,2. – функций main.

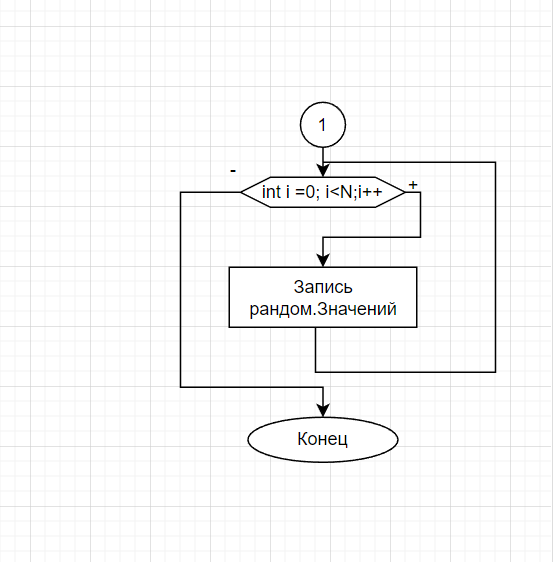


Рисунок 3. – заполнение массива.

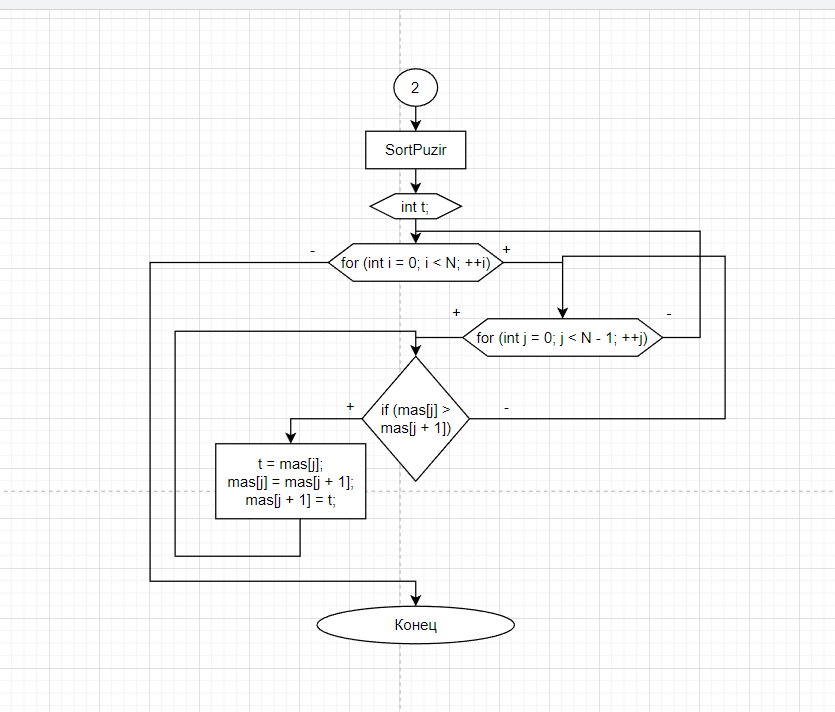


Рисунок 4. – Сортировка Пузырьком.

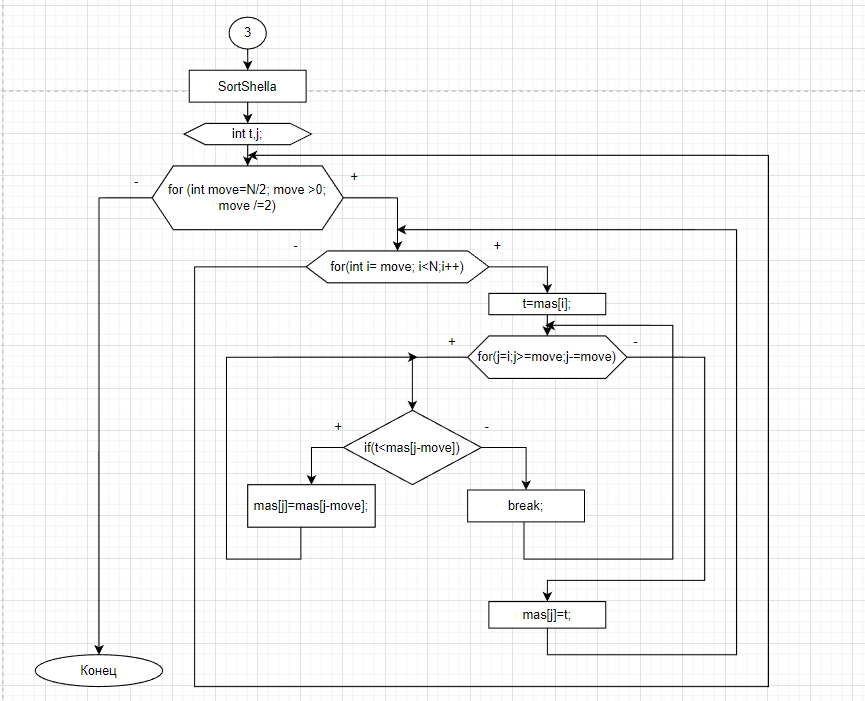
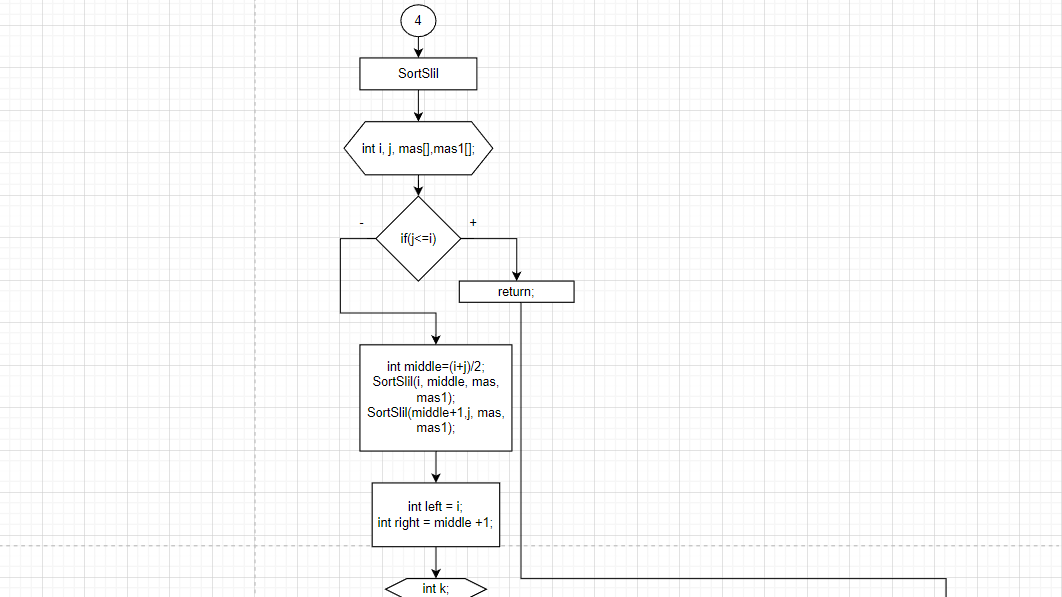
****

Рисунок 5. – Сортировка Шелла.

****

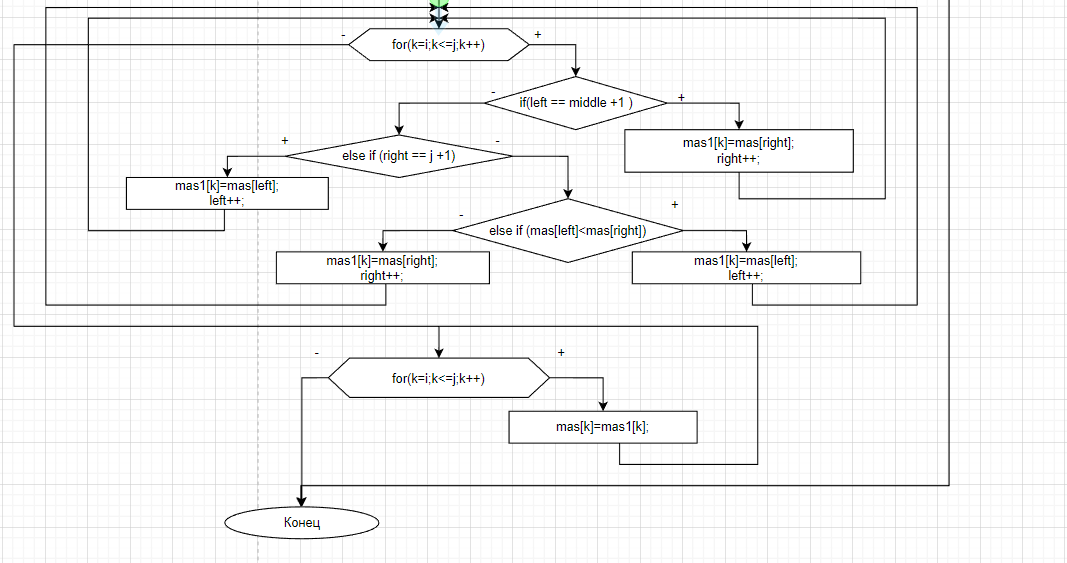
****

Рисунок 6, 7. – Сортировка слиянием.

**Реализация программы:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <Windows.h>

#include <time.h>

const int N = 22222;

int mas[N];

int mas1[N];

void SortPuzir()

{

int t;

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

{

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

{

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

{

t = mas[j];

mas[j] = mas[j + 1];

mas[j + 1] = t;

}

}

}

}

void SortShella()

{

int t, j;

for (int move = N / 2; move > 0; move /= 2)

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

{

t = mas[i];

for (j = i; j >= move; j -= move)

{

if (t < mas[j - move])

mas[j] = mas[j - move];

else

break;

}

mas[j] = t;

}

}

void SortSlil(int i, int j, int mas[], int mas1[]) {

if (j <= i) {

return;

}

int middle = (i + j) / 2;

SortSlil(i, middle , mas, mas1);

SortSlil(middle + 1, j, mas, mas1);

int left = i;

int right = middle + 1;

int k;

for (k = i; k <= j; k++) {

if (left == middle + 1) {

mas1[k] = mas[right];

right++;

}

else if (right == j + 1) {

mas1[k] = mas[left];

left++;

}

else if (mas[left] < mas[right]) {

mas1[k] = mas[left];

left++;

}

else {

mas1[k] = mas[right];

right++;

}

}

for (k = i; k <= j; k++) {

mas[k] = mas1[k];

}

}

void Random(FILE\* f)

{

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

{

fprintf\_s(f, "%d ", 0 + rand() % N);

}

}

int main()

{

FILE\* f = fopen("input.txt", "w");

if (f == NULL)

{

printf\_s("Файл не открылся!");

return 0;

}

Random(f);

fclose(f);

FILE\* f1 = fopen("input.txt", "r");

if (f1 == NULL)

{

printf\_s("Файл не открылся!");

return 0;

}

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

{

fscanf\_s(f1, "%d", &mas[i]);

}

clock\_t start1 = clock();

SortPuzir();

clock\_t end1 = clock();

double seconds1 = (double)(end1 - start1) / CLOCKS\_PER\_SEC;

clock\_t start2 = clock();

SortShella();

clock\_t end2 = clock();

double seconds2 = (double)(end2 - start2) / CLOCKS\_PER\_SEC;

clock\_t start3 = clock();

SortSlil(0, N - 1, mas, mas1);

clock\_t end3 = clock();

double seconds3 = (double)(end3 - start3) / CLOCKS\_PER\_SEC;

fclose(f1);

FILE\* f2 = fopen("output.txt", "w");

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

{

fprintf(f2, "%d ", mas[i]);

}

fclose(f2);

printf("Time puzir - %f\n", seconds1);

printf("Time shell - %f\n", seconds2);

printf("Time slil - %f\n", seconds3);

}

**Результат работы программы.**

Таблица 1. – время работы сортировок.

|  |  |  |  |
| --- | --- | --- | --- |
| Sort | Пузырёк, сек | Шелла, сек | Слияние, сек |
| 22222 | 3,7800 | 0,00200 | 0,00400 |
| 50000 | 19.91700 | 0,00500 | 0,01100 |

**Вывод:** ознакомился с сортировкой данных и ее видами.Самые эффективные сортировки это Шелла и слияние. Сортировка слиянием оказалась самая сложная как в понимании, так и в реализации. Сортировка Пузырьком проигрывает. Программа работает долго.