

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Утвержден на заседании кафедры

«Вычислительная техника»

" ____ " _____ 20 ____ г.

Заведующий кафедрой

_____ М.А. Митрохин

ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ: ОЗНАКОМИТЕЛЬНОЙ ПРАКТИКЕ
(2022/2023 учебный год)

_____ Маслов Юрий Андреевич

Направление подготовки (специальность) 09.05.01 «Применение и эксплуатация автоматизированных систем специального назначения»

Наименование профиля подготовки (специализация) «Эксплуатация вычислительных машин, комплексов, систем и сетей»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 5 лет

Год обучения _____ 1 _____ семестр _____ 2 _____

Период прохождения практики с 29.06.2023 по 12.07.2023

Кафедра «Вычислительная техника»

Заведующий кафедрой д.т.н., профессор, Митрохин М.А.

(должность, ученая степень, ученое звание, Ф.И.О.)

Руководитель практики д.т.н., профессор, Митрохин М.А.

(должность, ученая степень, ученое звание)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Утвержден на заседании кафедры

«Вычислительная техника»

" ____ " _____ 20 ____ г.

Заведующий кафедрой

_____ М.А. Митрохин

**ИНДИВИДУАЛЬНЫЙ ПЛАН ПРОХОЖДЕНИЯ УЧЕБНОЙ ПРАКТИКИ:
ОЗНАКОМИТЕЛЬНОЙ ПРАКТИКИ**

(2022/2023 учебный год)

_____ Маслов Юрий Андреевич

Направление подготовки (специальность) 09.05.01 «Применение и эксплуатация
автоматизированных систем специального назначения»

Наименование профиля подготовки (специализация) «Эксплуатация вычислительных
машин, комплексов, систем и сетей»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 5 лет

Год обучения _____ 1 _____ семестр _____ 2 _____

Период прохождения практики с 29.06.2023 по 12.07.2023

Кафедра «Вычислительная техника»

Заведующий кафедрой д.т.н., профессор, Митрохин М.А.

(должность, ученая степень, ученое звание, Ф.И.О.)

Руководитель практики д.т.н., профессор, Митрохин М.А.

(должность, ученая степень, ученое звание)

№ п/п	Планируемая форма работы во время практики	Количество часов	Календарные сроки проведения работы	Подпись руководителя практики от вуза
1	Выбор темы и разработка индивидуального плана проведения работ	2	29.06.2023 - 29.06.2023	
2	Подбор и изучение материала по теме работы	15	30.06.2023 – 02.07.23	
3	Разработка алгоритма	43	02.07.23 – 06.07.23	
4	Описание алгоритма и программы	18	06.07.23 – 08.07.23	
5	Тестирование	5	08.07.23 – 08.07.23	
6	Получение и анализ результатов	10	08.07.23 – 10.07.23	
7	Оформление отчёта	15	10.07.23 – 12.07.2023	
	Общий объём часов	108		

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

ОТЧЁТ
О ПРОХОЖДЕНИИ УЧЕБНОЙ ПРАКТИКЕ: ОЗНАКОМИТЕЛЬНОЙ
ПРАКТИКЕ

(2022/2023 учебный год)

Маслов Юрий Андреевич

Направление подготовки (специальность) 09.05.01 «Применение и эксплуатация автоматизированных систем специального назначения»

Наименование профиля подготовки (специализация) «Эксплуатация вычислительных машин, комплексов, систем и сетей»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 5 лет

Год обучения 1 семестр 2

Период прохождения практики с 29.06.2023 по 12.07.2023

Кафедра «Вычислительная техника»

Маслов Ю.А. выполнял практическое задание «Сортировка Шелла». На первоначальном этапе были изучен и проанализирован алгоритм сортировки Шелла, был выбран метод решения и язык программирования С, на котором была написана программа сортировки массива методом Шелла. Также, осуществил подсчёт времени выполнения сортировки. Протестировал и отладил программу. Оформил отчёт.

Специалист Маслов Ю.А. _____ " ____ " _____ 2023 г.

Руководитель Митрохин М.А. _____ " ____ " _____ 2023 г.
практики

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

ОТЗЫВ

ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ: ОЗНАКОМИТЕЛЬНОЙ ПРАКТИКЕ

(2022/2023 учебный год)

Маслов Юрий Андреевич

Направление подготовки (специальность) 09.05.01 «Применение и эксплуатация автоматизированных систем специального назначения»

Наименование профиля подготовки (специализация) «Эксплуатация вычислительных машин, комплексов, систем и сетей»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 5 лет

Год обучения 1 семестр 2

Период прохождения практики с 29.06.2023 по 12.07.2023

Кафедра «Вычислительная техника»

В процессе выполнения практики Маслов Ю.А. решал следующие задачи: разработка алгоритма сортировки Шелла, анализ работы алгоритма, сравнение существующих методов с разработанным алгоритмом.

За период выполнения практики были освоены основные понятия и технологии сортировки Шелла, реализованы подсчёт времени выполнения сортировки и тестирование с отладкой программы. Во время выполнения работы Маслов Ю.А. показал себя ответственным, добросовестным учеником, знающим свой предмет, имеющим представление о современном состоянии науки, владеющим современными общенаучными знаниями по информатике и вычислительной технике, программированию и сортировке.

За выполнение работы Маслов Ю.А. заслуживает оценки «_____».

Руководитель практики д.т.н., Митрохин М.А. « » _____ 2023 г.

Содержание

Введение.....	2
1 Постановка задачи.....	3
1.1 Достоинства алгоритма сортировки вставками.....	3
1.2 Недостатки алгоритма сортировки вставками.....	3
1.3 Типичные сценарии применения данного алгоритма.....	3
2 Выбор решения.....	4
3 Описание программы.....	5
4. Схемы программы.....	7
4.1 Блок-схема программы.....	7
4.2 Блок-схема алгоритма	8
5 Тестирование программы.....	9
6 Отладка.....	11
7 Совместная разработка	12
Заключение	14
Список используемой литературы	15
Приложение А. Результаты тестирования программы.....	16
Приложение Б. Листинг программы	18

Введение

Сортировка данных на сегодняшний день при современном развитии компьютерных технологий является одним из наиболее распространенных процессов современной обработки данных. Задачи на сортировку данных встречаются очень часто в различных профессиональных сферах деятельности.

Алгоритмы сортировки образуют отдельный класс алгоритмов, применяются практически во всех задачах обработки информации. При этом они настолько тесно связаны друг с другом, что образуют отдельный класс алгоритмов. Алгоритмы сортировки, как правило, применяются с целью осуществления последующего более быстрого поиска. Например, трудно пользоваться словарями, если бы слова в них не были бы упорядочены по алфавиту.

Важность сортировки основана на том факте, что на ее примере можно показать многие основные фундаментальные приемы и методы построения алгоритмов. Сортировка является хорошим примером огромного разнообразия алгоритмов, которые выполняют одну и ту же задачу. Кроме того, многие из них имеют определенные преимущества друг перед другом. За счет усложнения алгоритма можно добиться существенного увеличения эффективности и быстродействия алгоритма по сравнению с более простыми методами. Как правило, термин сортировка понимают, как процесс перестановки объектов некоторого множества в определенном порядке.

Сортировка Шелла — алгоритм сортировки, являющийся усовершенствованным вариантом сортировки вставками. Идея метода Шелла состоит в сравнении элементов, стоящих не только рядом, но и на определенном расстоянии друг от друга.

1 Постановка задачи

Поставленная задача: необходимо заполнить массив из n -ого количества элементов случайными числами, записать данные элементы в отдельный файл. После этого выполнить сортировку Шелла над данными, находящимися в массиве, записать отсортированные данные в другой файл, посчитать время выполнения алгоритма при сортировке.

Необходимо использовать сервис GitHub для совместной работы, создать и выложить коммиты, характеризующие действия, выполненные каждым участником бригады.

Оформить отчет по проведенной практике.

1.1 Достоинства алгоритма сортировки вставками

- отсутствие потребности в памяти под стек;
- отсутствие деградации при неудачных наборах данных — быстрая сортировка легко деградирует до $O(n^2)$, что хуже, чем худшее гарантированное время для сортировки Шелла;

1.2 Недостатки алгоритма сортировки вставками

- неизвестен оптимальный набор расстояний;
- высокая алгоритмическая сложность $O(n^2)$;

1.3 Типичные сценарии применения данного алгоритма

- товары в магазине (сортировка по цене, году выпуска, габаритам, весу, срокам поставки);
- студенты в вузе (сортировка по среднему балу, кол-ву прогулов, уровню IQ, числу хвостов, ФИО);
- города/страны (сортировка по населению, рождаемости, ВВП, ВВП на душу населения);

2 Выбор решения

Нашей бригадой было выбрано вести разработку в среде Microsoft Visual Studio на языке C.

Для написания данной программы будет использован язык программирования Си. Этот язык является распространённым языком программирования. При разработке языка Си был принят компромисс между низким уровнем языка ассемблера и высоким уровнем других языков. Си – это язык программирования общего назначения, хорошо известный своей эффективностью, экономичностью и переносимостью. Указанные преимущества Си обеспечивают хорошее качество разработки почти любого вида программного продукта.

Microsoft Visual Studio — это программная среда по разработке приложений для ОС Windows, как консольных, так и с графическим интерфейсом.

3 Описание программы

При запуске программы выводится сообщение, в котором пользователю необходимо ввести количество значений для сортировки

```
printf("Введите количество элементов массива: ");  
scanf("%d", &size);
```

После того, как данные были введены, генерируется массив из случайных чисел, эти числа записываются в файл input.txt.

```
f = fopen("input.txt", "w");  
for (int i = 0; i < size; i++)  
{  
    array[i] = rand();  
    fprintf(f, "%d ", array[i]);  
}  
fclose(f);
```

Далее над этими данными выполняется сортировка Шелла, при которой сначала сравниваются и сортируются между собой значения, стоящие один от другого на некотором расстоянии d . После этого процедура повторяется для некоторых меньших значений d , а завершается упорядочиванием элементов при $d = 1$ (то есть обычной сортировкой вставками).

```
void shellSort(int array[], int n) {  
    // Вычисляем интервал  
    int interval = 5;  
    while (interval < n / 3) {  
        interval = interval * 3 + 1;  
    }  
    // Применяем сортировку Shell  
    while (interval > 0) {  
        for (int i = interval; i < n; i++) {  
            int temp = array[i];  
            int j = i;  
            // Сортируем подмассивы с интервалом interval  
            while (j >= interval && array[j - interval] > temp) {
```

```

        array[j] = array[j - interval];
        j -= interval;
    }
    array[j] = temp;
}
interval = (interval - 1) / 3;
}

```

После этого отсортированный массив записывается в файл output.txt

```

f = fopen("output.txt", "w");
for (int i = 0; i < size; i++) {
    fprintf(f, "%d ", array[i]);
}
fclose(f);

```

Подробный алгоритм работы программы и функции сортировки представлен в разделе 4 на рисунках 1, 2.

Листинг программы приведен в приложении А.

4. Схемы программы

4.1 Блок-схема программы

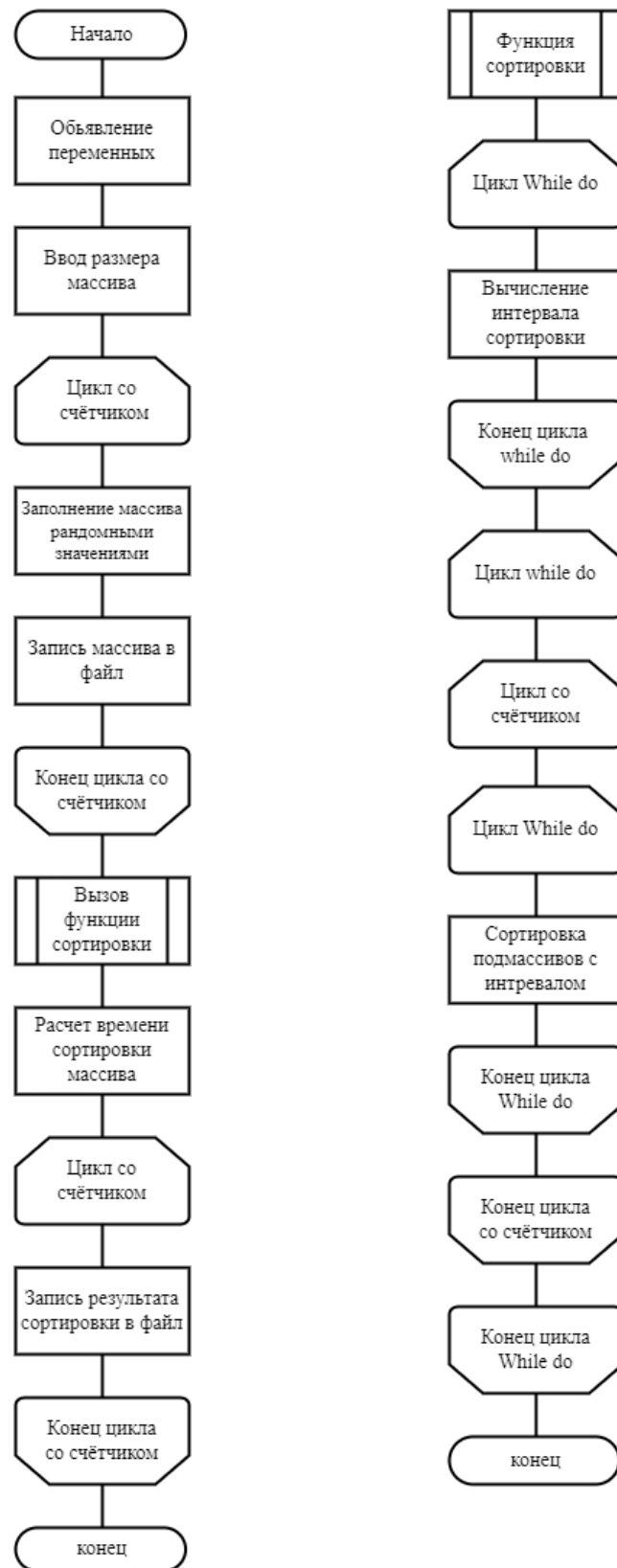


Рисунок 1 - Блок-схема программы

4.2 Блок-схема алгоритма

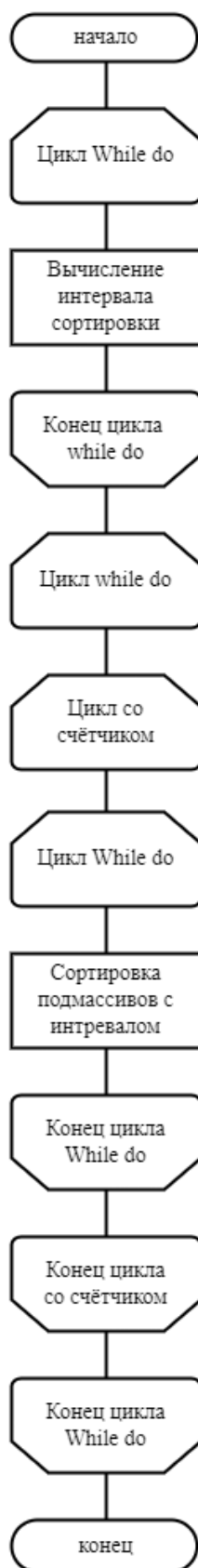


Рисунок 2 - Блок-схема алгоритма с подключенными файлами

5 Тестирование программы

5.1 Тестирование на разных наборах данных

Тестовый набор данных представлен в таблице 1. Результаты тестирования приведены в Приложении А на рисунках А.1 - А.9.

Таблица 1 – Тестовый набор данных

№	Размера массива	Время выполнения сортировки в секундах
1	100000	0,015
2	500000	0,096
3	1000000	0,211
4	1500000	0,331
5	2000000	0,457
6	2500000	0,581
7	3000000	0,736
8	3500000	0,86
9	4000000	1,02

5.2 Анализ полученных результатов тестирования (анализ работы алгоритма)

На основании анализа данных, полученных в результате тестирования алгоритма сортировки вставками, можно сделать вывод, что время, затраченное на работу программы относительно количества элементов, увеличивается линейно, то есть с увеличением количества элементов пропорционально увеличивается время работы программы.

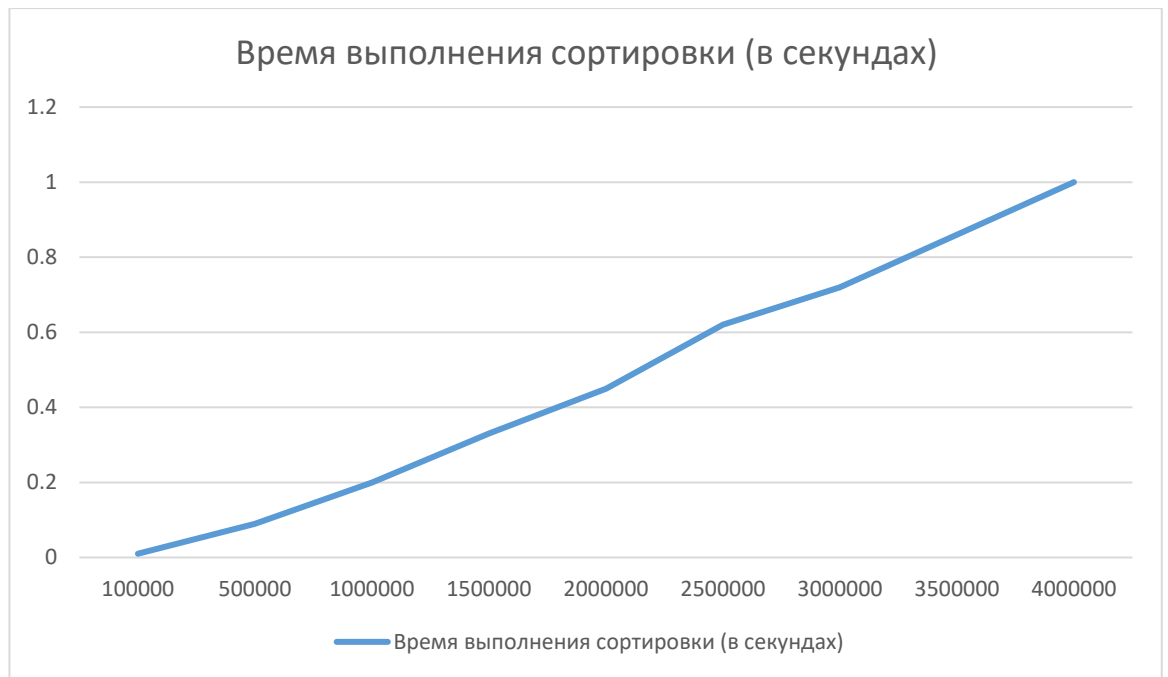


Рисунок 3 – Результаты тестирования

6 Отладка

В качестве среды разработки была выбрана программа Microsoft Visual Studio, которая содержит в себе все необходимые средства для разработки и отладки модулей и программ.

Для отладки программы использовались точки останова и пошаговое выполнение кода программы, анализ содержимого локальных переменных.

Точки останова – это прерывание выполнения программы, при котором выполняется вызов отладчика. Отладчик является инструментом для поиска и устранения ошибок в программе, с помощью которого можно исследовать состояние программы.

Был использован метод бинарного поиска, он включает в себя разделение частей кода для упрощения процесса отладки. Это может быть особенно полезно, если причина ошибки находится в начале языка программирования, а фактическая ошибка ближе к концу.

Команда шаг с заходом (step into) выполняет следующую инструкцию в обычном пути выполнения программы, а затем приостанавливает выполнение программы, чтобы мы могли проверить состояние программы с помощью отладчика. Если выполняемый оператор содержит вызов функции, шаг с заходом заставляет программу перескакивать в начало вызываемой функции, где она приостанавливается.

Тестирование проводилось в рабочем порядке, в процессе разработки, после завершения написания программы. После завершения написания программы, мною были выявлены и исправлены ошибки.

7 Совместная разработка

Во время работы над данной практикой наша бригада осуществляла совместную работу в GitHub.

Мною был разработан интерфейс программы, реализованы подсчёт времени выполнения сортировки и тестирование с отладкой программы, это было зафиксировано и загружено на удаленный репозиторий GitHub, на ветку main.

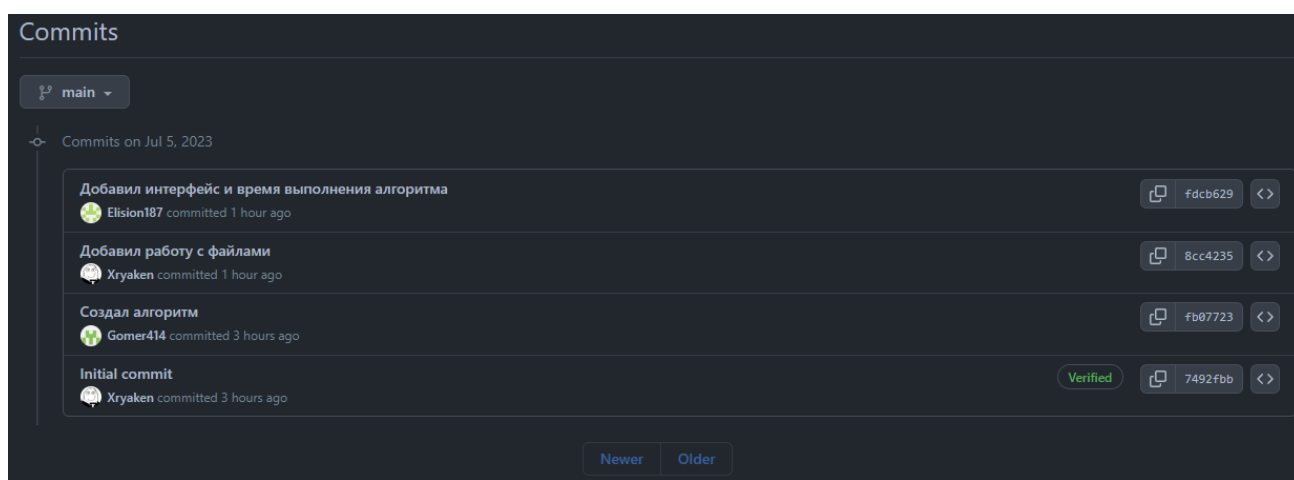


Рисунок 4 – Созданные коммиты

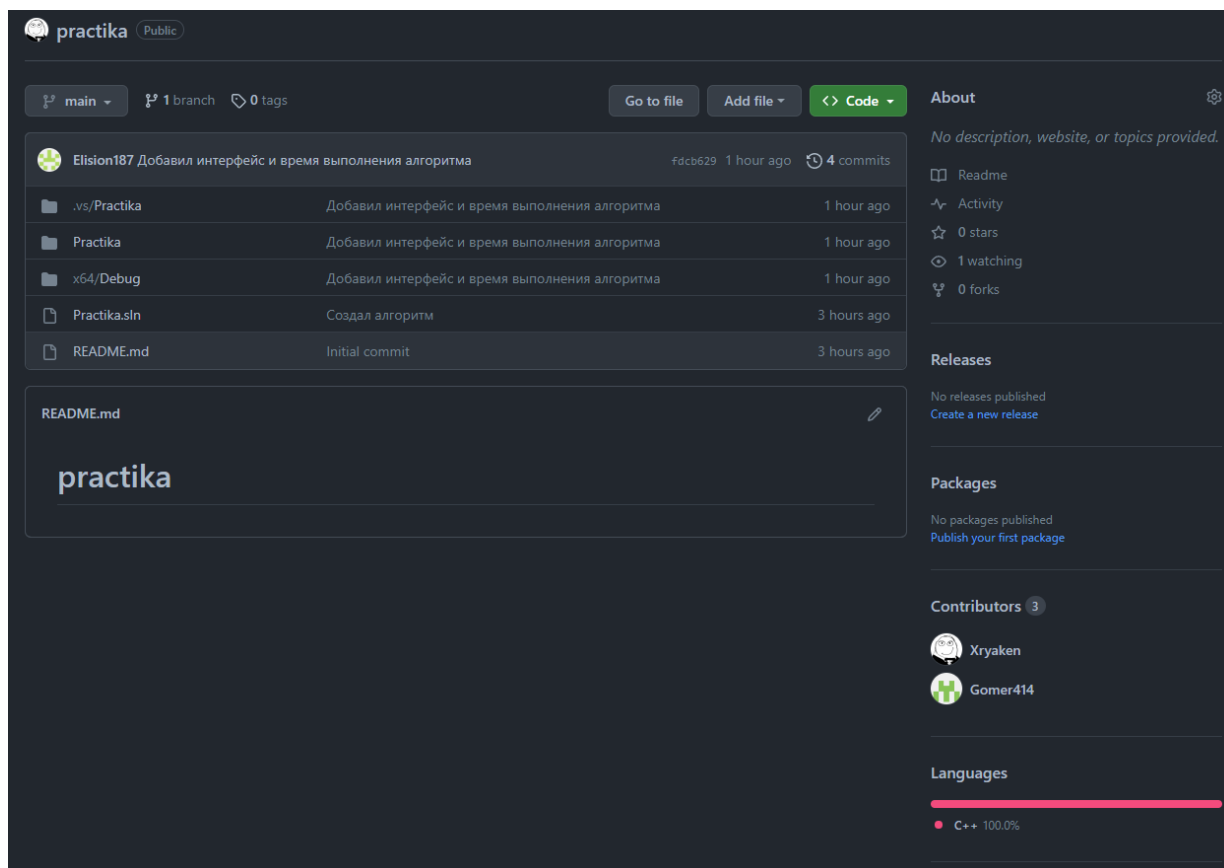


Рисунок 5– Ветка main

Для загрузки данных на локальный репозиторий, а также отправки данных на удаленный репозиторий было использовано приложение GitHub Desktop.

Ссылка на удаленный репозиторий:

<https://github.com/Xryaken/practika.git>

Заключение

При выполнении данной работы были получены навыки совместной работы с помощью сервисов GitHub и GitHub Desktop. Был изучен алгоритм сортировки Шелла.

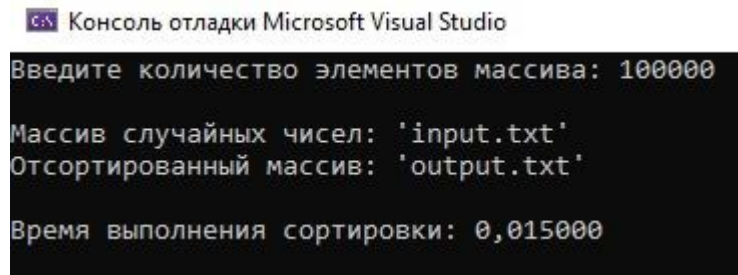
Мною было осуществлен подсчёт времени выполнения сортировки. А также тестирование и отладка программы. При выполнении практической работы были улучшены базовые навыки программирования на языке C. Улучшены навыки отладки, тестирования программ и работы со сложными типами данных.

В дальнейшем программу можно улучшить путем подключения упрощающих реализацию данной сортировки библиотек и улучшения графического интерфейса.

Список используемой литературы

1. ГОСТ 19.701 – 90 Схемы алгоритмов, программ, данных и систем.
2. Керниган, Брайан У., Ритчи, Деннис М. Язык программирования С, 2-е издание.: Пер. с англ. – М., 2009.
3. ShellSort [Электронный ресурс] – URL:
<https://www.geeksforgeeks.org/shellsort/> (дата обращения: 04.07.2023 г)

Приложение А. Результаты тестирования программы



Консоль отладки Microsoft Visual Studio

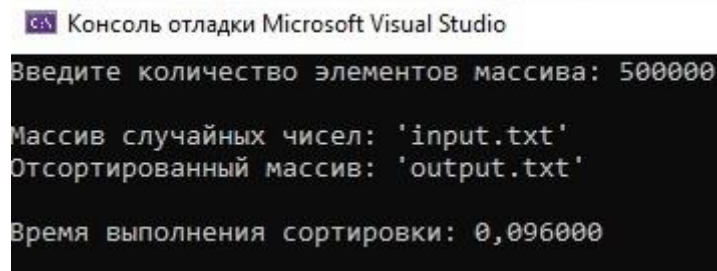
Введите количество элементов массива: 100000

Массив случайных чисел: 'input.txt'

Отсортированный массив: 'output.txt'

Время выполнения сортировки: 0,015000

Рисунок А.1.



Консоль отладки Microsoft Visual Studio

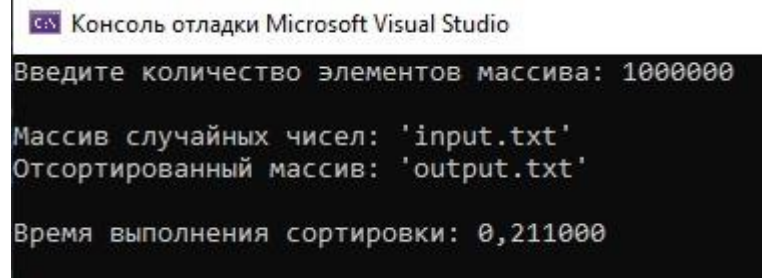
Введите количество элементов массива: 500000

Массив случайных чисел: 'input.txt'

Отсортированный массив: 'output.txt'

Время выполнения сортировки: 0,096000

Рисунок А.2.



Консоль отладки Microsoft Visual Studio

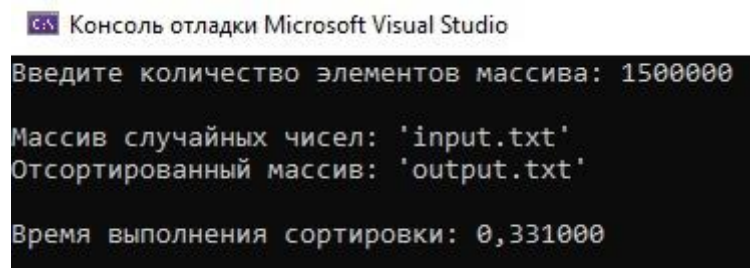
Введите количество элементов массива: 1000000

Массив случайных чисел: 'input.txt'

Отсортированный массив: 'output.txt'

Время выполнения сортировки: 0,211000

Рисунок А.3.



Консоль отладки Microsoft Visual Studio

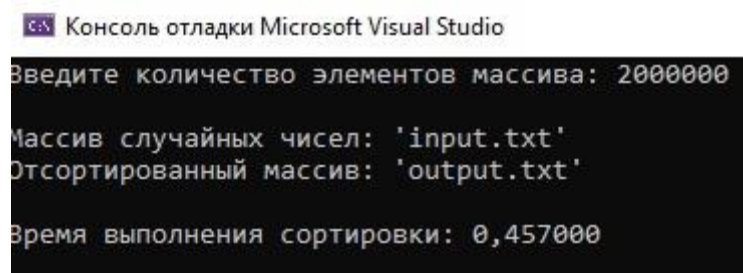
Введите количество элементов массива: 1500000

Массив случайных чисел: 'input.txt'

Отсортированный массив: 'output.txt'

Время выполнения сортировки: 0,331000

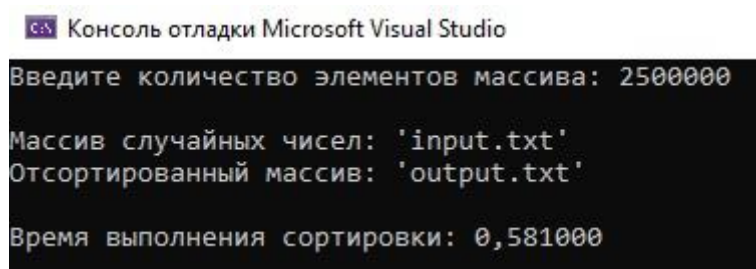
Рисунок А.4.



Консоль отладки Microsoft Visual Studio

```
Введите количество элементов массива: 2000000  
Массив случайных чисел: 'input.txt'  
Отсортированный массив: 'output.txt'  
Время выполнения сортировки: 0,457000
```

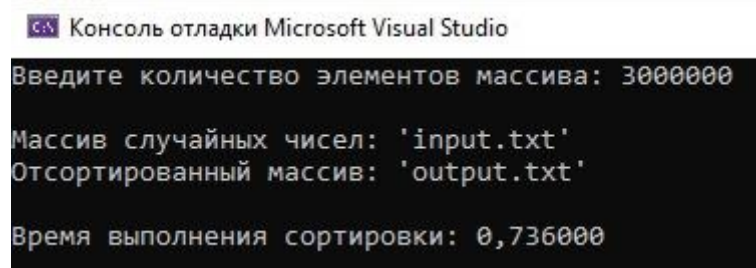
Рисунок А.5.



Консоль отладки Microsoft Visual Studio

```
Введите количество элементов массива: 2500000  
Массив случайных чисел: 'input.txt'  
Отсортированный массив: 'output.txt'  
Время выполнения сортировки: 0,581000
```

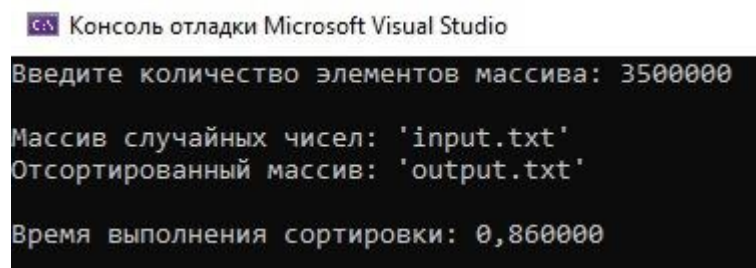
Рисунок А.6.



Консоль отладки Microsoft Visual Studio

```
Введите количество элементов массива: 3000000  
Массив случайных чисел: 'input.txt'  
Отсортированный массив: 'output.txt'  
Время выполнения сортировки: 0,736000
```

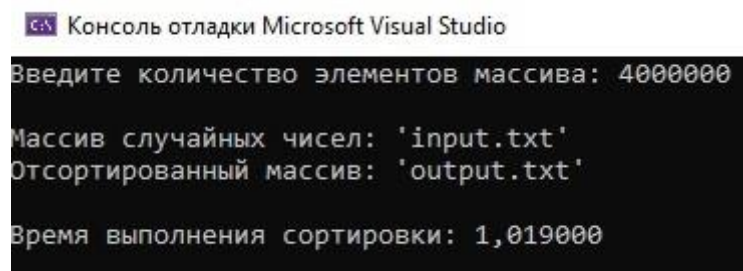
Рисунок А.7.



Консоль отладки Microsoft Visual Studio

```
Введите количество элементов массива: 3500000  
Массив случайных чисел: 'input.txt'  
Отсортированный массив: 'output.txt'  
Время выполнения сортировки: 0,860000
```

Рисунок А.8.



Консоль отладки Microsoft Visual Studio

```
Введите количество элементов массива: 4000000  
Массив случайных чисел: 'input.txt'  
Отсортированный массив: 'output.txt'  
Время выполнения сортировки: 1,019000
```

Рисунок А.9.

Приложение Б. Листинг программы

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <locale.h>

void shellSort(int array[], int n) {
    // Вычисляем интервал
    int interval = 5;
    while (interval < n / 3) {
        interval = interval * 3 + 1;
    }

    // Применяем сортировку Shell
    while (interval > 0) {
        for (int i = interval; i < n; i++) {
            int temp = array[i];
            int j = i;

            // Сортируем подмассивы с интервалом interval
            while (j >= interval && array[j - interval] > temp)
            {
                array[j] = array[j - interval];
                j -= interval;
            }

            array[j] = temp;
        }

        interval = (interval - 1) / 3;
    }
}
```

```

    }
}

int main() {
    FILE* f;
    int size;
    int* array;
    srand(time(NULL));
    setlocale(LC_ALL, "Rus");

    printf("Введите количество элементов массива: ");
    scanf("%d", &size);
    printf("\n");

    array = (int*)malloc(size * sizeof(int));

    printf("Массив случайных чисел: 'input.txt'\n");
    f = fopen("input.txt", "w");

    // Генерируем случайные значения
    for (int i = 0; i < size; i++)
    {
        array[i] = rand();
        fprintf(f, "%d ", array[i]);
    }
    fclose(f);

    printf("Отсортированный массив: 'output.txt'\n");
    f = fopen("output.txt", "w");

    time_t start = clock();

```



```

// Сортируем массив с помощью сортировки Shell
shellSort(array, size);

time_t stop = clock();           //время после сортировки
for (int i = 0; i < size; i++) {
    fprintf(f, "%d ", array[i]);
}
fclose(f);
double time = (stop - start) / 1000.0;    //время сортировки
printf("\n");
printf("Время выполнения сортировки: %lf\n", time);

return 0;
}

```