Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное   
учреждение высшего образования

Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

**Отчет по лабораторной работе**

**«Сортировка слиянием»**

**Выполнил**:

студент группы 382003-1

Филатов Андрей Александрович

**Проверил**:

ассистент каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2020

**Содержание**

[Постановка задачи 3](#_Toc26962562)

[Метод решения 4](#_Toc26962563)

[Руководство пользователя 5](#_Toc26962564)

[Описание программной реализации 6](#_Toc26962565)

[Подтверждение корректности 7](#_Toc26962566)

[Результаты экспериментов 8](#_Toc26962567)

[Заключение 9](#_Toc26962568)

[Приложение 10](#_Toc26962569)

# Постановка задачи

Написать на языке С сортировку слиянием (Merge *sort*).

# Метод решения

Алгоритм сортировки слиянием:

1. Если в рассматриваемом массиве один элемент, то он уже отсортирован — алгоритм завершает работу.
2. Иначе массив рекурсивно разбивается пополам, и каждая из половин делиться до тех пор, пока размер очередного подмассива не станет равным единице;
3. После сортировки двух частей массива к ним применяется процедура слияния, которая по двум отсортированным частям получает исходный отсортированный массив.

Алгоритм слияния:

1. Сравниваем элементы массивов (начиная с начала) и меньший из них записываем в финальный.
2. В массиве, у которого оказался меньший элемент, переходим к следующему элементу и сравниваем теперь его.
3. В конце, если один из массивов закончился, мы просто дописываем в финальный оставшийся массив.

Посчитаем сложность данного алгоритма. Пусть список содержит n элементов.

Тогда за его можно разделить на две части и после сортировки слить их вместе. Каждая из этих двух частей имеет размер n/2, и за  шагов каждую из них можно поделить на две части размером n/4 и затем после сортировки слить их вместе. Аналогично, четыре части размером n/4 за суммарное  шагов делятся на части размером n/8 и сливаются вместе. Этот процесс «в глубину» продолжается столько раз, сколько раз можно число n делить на 2, до тех пор, пока размер части не станет равен 1, то есть . Итого, общая сложность этого алгоритма равна

Одним из главных недостатков сортировки слиянием то, что данный алгоритм требует дополнительную память для работы – дополнительная память .

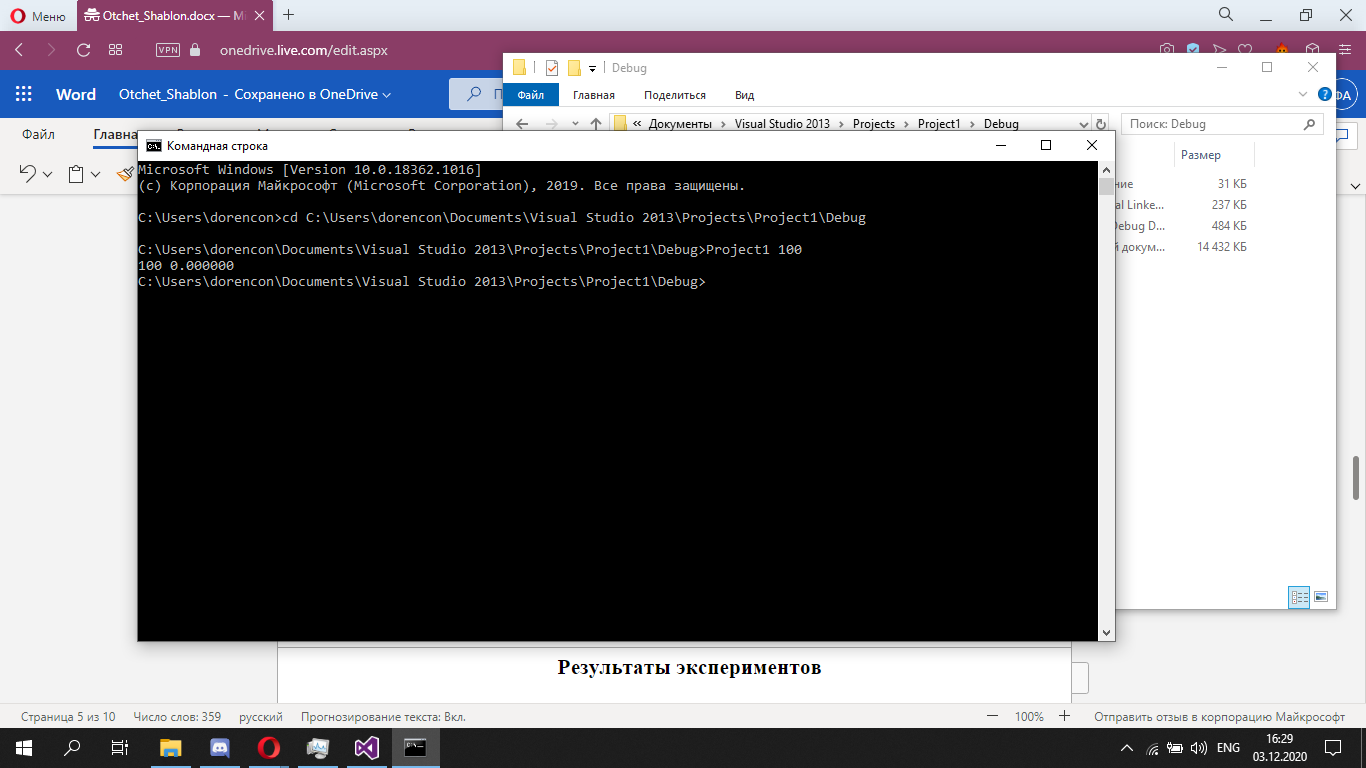
Количество обменов, необходимое для выполнения сортировки массива .

# Руководство пользователя

Для использования программы пользователь должен иметь файл с именем test.txt, содержащий неотсортированный массив.

Для удобства тестирования, программа запускается из командной строки с одним аргументом. Аргумент – сколько последовательных чисел прочитать из файла.

Пример командной строки



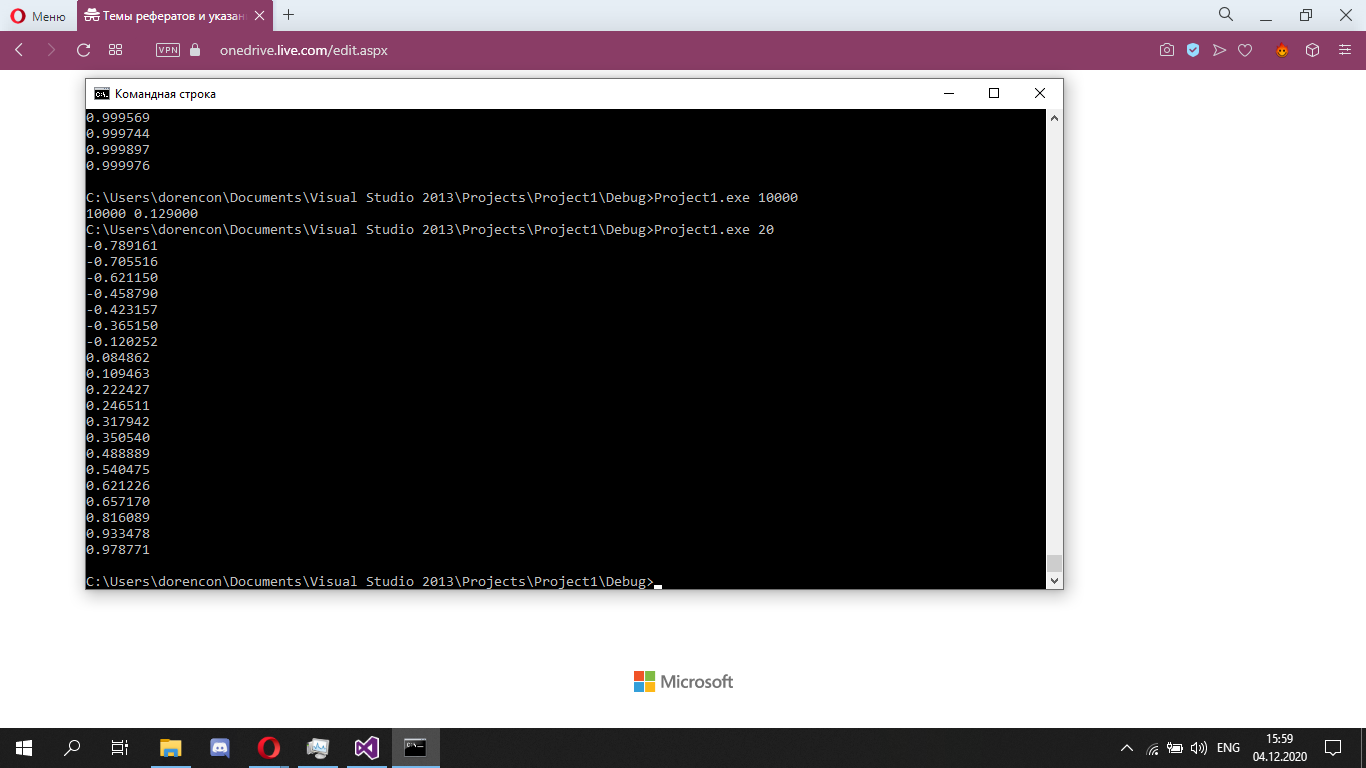
Результат работы программы – отсортированный массив, будет выведен на экран.

# Описание программной реализации

Программа состоит из одного текстового (тестового) файла и исполняемого файла.

# Подтверждение корректности

Для подтверждения корректности в программе были произведены тесты с различным количеством элементов. Пример работы программы на 20 элементах.



# Результаты экспериментов

По данным экспериментов видно, что сортировка слиянием — логарифмический алгоритм, время его работы пропорционально произведению размера сортируемого списка на его логарифм.

Тесты проводились на ноутбуке Aser Intel Core i5-7200U CPU 2.50 GHz

# Заключение

Сортировка слиянием – это сортировка, работающая по принципу “Разделяй и властвуй”. Алгоритм был изобретён [Джоном фон Нейманом](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D0%BD_%D0%9D%D0%B5%D0%B9%D0%BC%D0%B0%D0%BD,_%D0%94%D0%B6%D0%BE%D0%BD) в [1945 году](https://ru.wikipedia.org/wiki/1945_%D0%B3%D0%BE%D0%B4). Достаточно быстрый алгоритм,

общая сложность этого алгоритма равна

Одним из главных недостатков сортировки слиянием то, что данный алгоритм требует дополнительную память для работы – дополнительная память . Однако, существуют модифицированные алгоритмы слияния, которые при тех же временных затратах, требуют только (1 дополнительной памяти.

# Приложение

#if defined(\_WIN32) || defined(\_\_WIN32\_\_) || defined(WIN32)

#define \_CRT\_SECURE\_NO\_WARNINGS

#pragma warning(disable:4996)

#endif

#include <stdio.h>

#include <malloc.h>

#include <string.h>

#include <time.h>

void merge(double\* a, double\* b, int l, int r)

{

if (l >= r)

{

return;

}

int m = (l + r) / 2;

merge(b, a, l, m);

merge(b, a, m + 1, r);

int i = l, j = m + 1;

while ((i <= m) || (j <= r))

{

if ((j <= r) && ((i > m) || (b[i] > b[j])))

{

a[i + j - m - 1] = b[j];

j++;

}

else

{

a[i + j - m - 1] = b[i];

i++;

}

}

return;

}

int main(int argc, char \*\*argv)

{

int n = atoi(argv[1]);

FILE \*f;

f = fopen("test.txt", "r");

double \*a, \*b;

a = malloc(n \* sizeof(double));

b = malloc(n \* sizeof(double));

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

{

fscanf(f, "%lf", &a[i]);

b[i] = a[i];

}

int t0 = clock();

merge(a, b, 0, n - 1);

int t1 = clock();

/\*for (int i = 0; i < n; i++)

{

printf("%f\n", a[i]);

}\*/

printf("%d %lf", n, (double)(t1 - t0) / CLOCKS\_PER\_SEC);

return 0;

}