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 выбором»**

**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 выбором (*Selection sort*).

# Метод решения

Для сортировки выбором проходим по массиву в поисках минимального элемента. Найденный минимум меняем местами с первым элементом. Неотсортированная часть массива уменьшилась на один элемент (не включает первый элемент, куда переставили найденный минимум). К этой неотсортированной части применяем те же действия — находим минимум и ставим его на первое место в неотсортированной части массива. И так продолжаем до тех пор, пока неотсортированная часть массива не уменьшится до одного элемента.

Посчитаем сложность данного алгоритма. Пусть список содержит n элементов. Сначала нужно найти минимум среди n элементов списка, что потребует n операций. Потом нужно найти наименьший из n-1 элемента, на это нужно n-1 операция. Потом нужно n-2 операции и т. д. Общее число операций равно

Таким образом, сортировка выбором — квадратичный алгоритм, время его работы пропорционально квадрату от размера списка.

Сортировка выбором не требует дополнительной памяти для работы – дополнительная память .

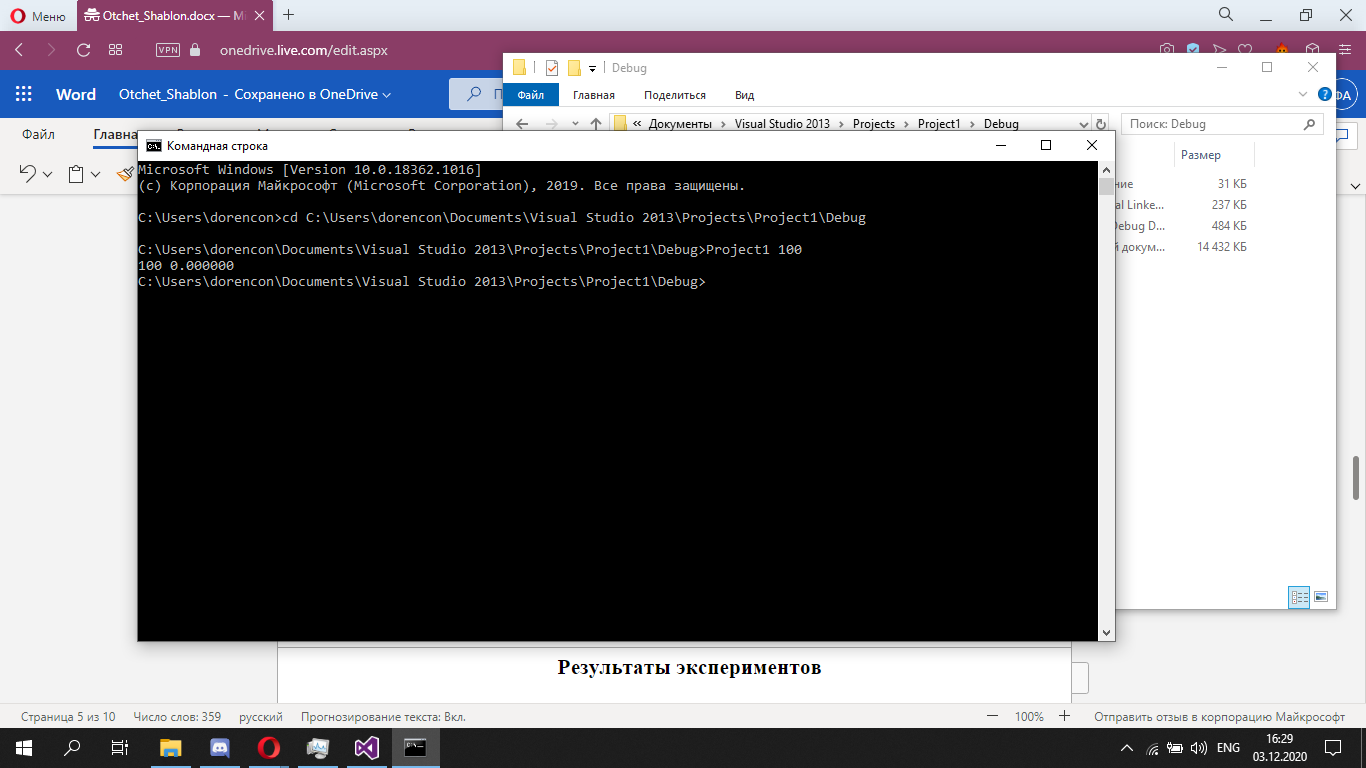
Среднее количество обменов, необходимое для выполнения сортировки массива .

# 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



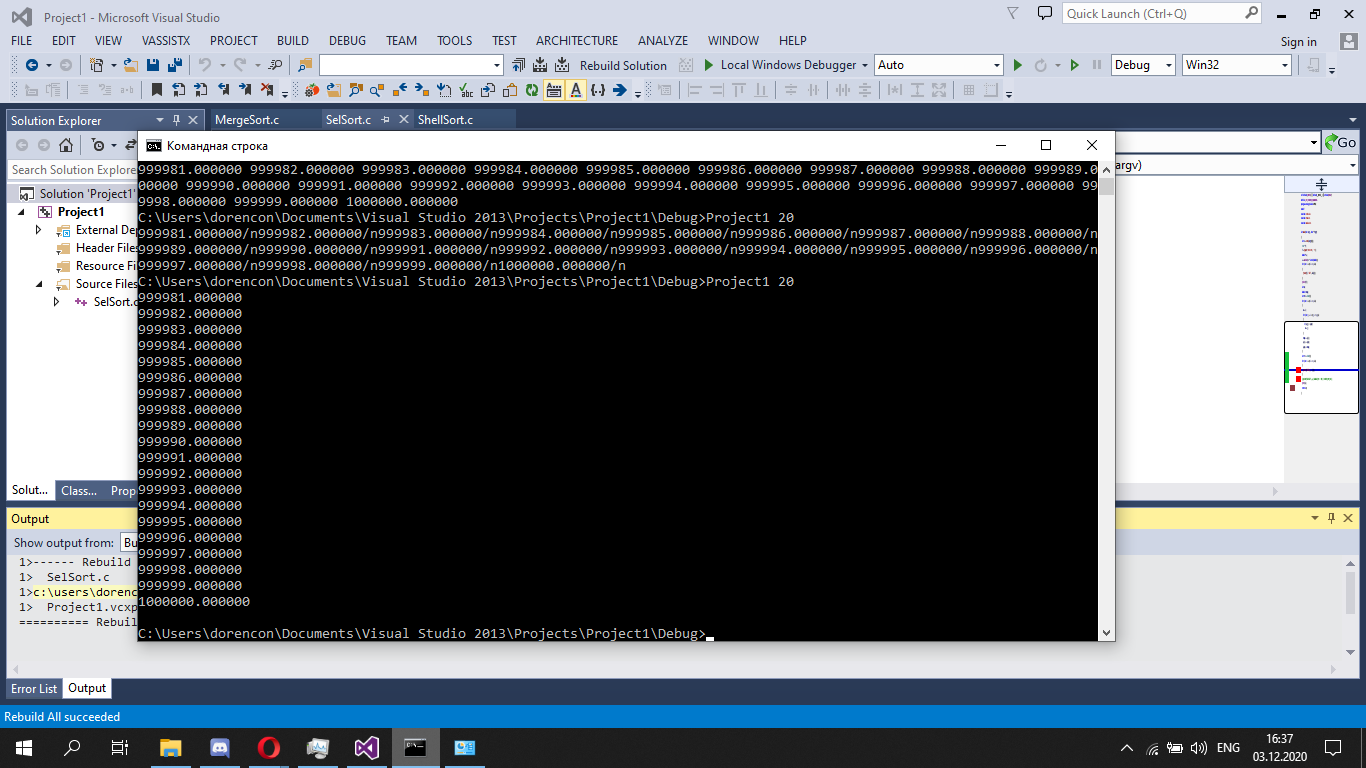
Результат работы программы – отсортированный массив, будет выведен на экран.

# 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 выбором — квадратичный алгоритм, время его работы пропорционально квадрату от размера сортируемого списка.

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

# Заключение

Сортировка выбором – это простая в реализации сортировка, но на больших массивах данных будет работать очень медленно. Это ограничивает его применимость в реальных задачах.

# Приложение

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

#define \_CRT\_SECURE\_NO\_WARNINGS

#pragma warning(disable:4996)

#endif

#include <stdio.h>

#include <time.h>

#include <malloc.h>

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

{

int n = atoi(argv[1]);

FILE \*f;

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

double\* a;

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

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

{

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

}

fclose(f);

int m;

double temp;

int t0 = clock();

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

{

m = i;

for (int j = i + 1; j < n; j++)

{

if (a[j] < a[m])

m = j;

}

temp = a[i];

a[i] = a[m];

a[m] = temp;

}

int t1 = clock();

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

free(a);

return 0;

}