Федеральное агентство по образованию Российской Федерации

Государственное образовательное учреждение

высшего профессионального образования

Нижегородский государственный университет им. Н.И. Лобачевского

Институт Информационных Технологий Математики и Механики

Отчёт по лабораторной работе

Сравнение сортировок

Выполнила:

студентка ф-та ИТММ гр. 3821Б1ПМ3

Крошко Е. И.

Проверил:

заведующий лабораторией суперкомпьютерных технологий и высокопроизводительных вычислений

Лебедев И.Г.

Нижний Новгород

2021 г.

Оглавление

1. [Введение. 3](file:///C:\Users\Катя\Downloads\Отчет%20по%202ой%20лабе%20(1).docx#_Toc89791253)

[2. Постановка задачи. 4](file:///C:\Users\Катя\Downloads\Отчет%20по%202ой%20лабе%20(1).docx#_Toc89791254)

[3. Руководство пользователя. 5](file:///C:\Users\Катя\Downloads\Отчет%20по%202ой%20лабе%20(1).docx#_Toc89791255)

4. Руководство программиста. 7

5. Эксперименты………………………………………………………………………………....12

6. [Заключение. 13](file:///C:\Users\Катя\Downloads\Отчет%20по%202ой%20лабе%20(1).docx#_Toc89791257)

[Литература. 14](file:///C:\Users\Катя\Downloads\Отчет%20по%202ой%20лабе%20(1).docx#_Toc89791258)

[Приложение. 15](file:///C:\Users\Катя\Downloads\Отчет%20по%202ой%20лабе%20(1).docx#_Toc89791259)

[Приложение 1. 1](file:///C:\Users\Катя\Downloads\Отчет%20по%202ой%20лабе%20(1).docx#_Toc89791260)5

[Приложение 2. 16](file:///C:\Users\Катя\Downloads\Отчет%20по%202ой%20лабе%20(1).docx#_Toc89791261)

# **1.Введение**

Программирование — неотъемлемая часть нашей жизни. это возможность взаимодействия человека с машиной, возможность общения друг с другом на понятном языке. Всё, что мы делаем на компьютере, так или иначе связано с ними.

Методы сортировки — это нужная и интересная вещь во многих проектах и разработках. В данной лабораторной применяется 3 метода сортировок и опробовал их на случайных числах, записанных в файле.

# **2.Постановка задачи**

Реализовать сортировки массивов данных (тип данных определяется преподавателем) задаваемых: обязательно случайно, дополнительно с клавиатуры или из файла.

Реализовать сортировки: пузырьком, вставкой, быстрая. Сравнить время работы, сделать выводы.

Первая программа создает текстовый файл с записанными в него числами. Программа принимает количество чисел n, максимальное и минимальное значение.

Вторая программа читает текстовый файл с набором чисел, выводит консольный интерфейс (печать, сортировка, сброс, выход), выполняет выбранные действия.

# **3.Руководство пользователя.**

**3.1. Первая программа**

Используется две программы. После запуска первой программы выводится сообщение на экран: «Введите количество чисел n = » (см. Рисунок 1)

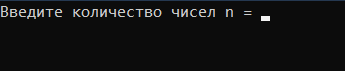


Рис.1. Консоль. Количество чисел

Далее пользователю предлагается ввести значения минимума и максимума в диапазоне от -128 до 127. (см. Рисунок 2)

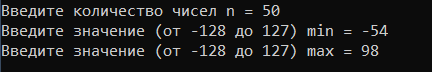


Рис.2. Консоль. Ввод максимального и минимального значений

При введение некорректного значения n (количество чисел) меньше или равного 0, будет выведено сообщение об ошибке. (см. Рисунок 3)

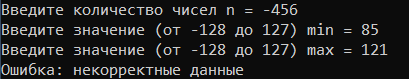


Рис.3. Консоль. Введены некорректные данные

Результатом работы программы будет текстовый файл, который наполнен некоторым количеством чисел. (см. Рисунок 4)

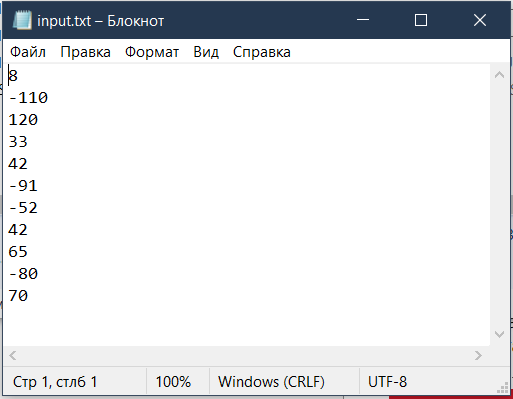


Рис. 4. Текстовый файл

Если в процессе работы программы вы введи некорректные данные, запустите программу заново, введите новые значения.

## **3.2. Вторая программа**

Далее нужно запустить вторую программу, где появится сообщение с набором команд, которые может выполнить программа: печать, сортировка, отчистка массива, завершение работы. (см. Рисунок 5)

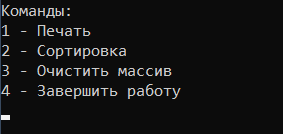


Рис. 5. Консоль с выбором команд

При вводе команды «2» программа предложит пользователю выбрать тип сортировки из тех, что были реализованы в программе. (см Рисунок 6)

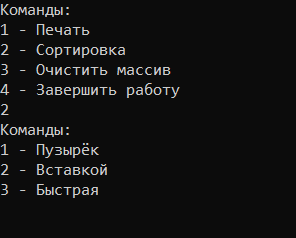


Рисунок 6. Выбор сортировки

После выбора типа сортировки программа отсортирует числа в файле, сообщит о завершении операции и укажет время, затраченное на сортировку в секундах. (см. Рисунок 7)

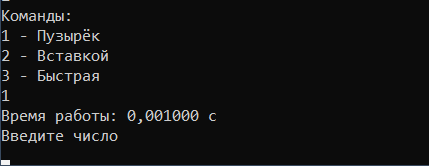


Рис.7. Выбор сортировки. Время работы

# **4.Руководство программиста.**

**4.1. Описание структуры программы.**

Первая программа состоит из одного модуля int main(), где находиться код создания чисел и записи их в файл. В нее входят такие функции, как printf (для вывода информации на экран), scanf (для ввода данных с клавиатуры), fprintf (для записи данных в тестовый файл), fclose (для закрытия текстового документа)

Вторая программа также состоит из одного модуля int main(), где находится код консоли. Используются переменные типа int (Рисунок. 8), FILE\* (Рисунок. 9) и char\*



Рис. 8. Объявление переменных типа int



Рис. 9. Объявление переменных типа FILE\*

Важный блок программы заключается в интерактивного меню, выполненного при помощи конструкций if и else, и ссылающихся на ранее объявленные функции. Также при помощи библиотеки <time.h>, используем функцию clock\_t clock(void), которая возвращает время, затраченное процессором на выполнение программы.

while (a != 4)

{

char\* mas = NULL;

FILE\* file = 0;

file = fopen("input.txt", "r");

fscanf\_s(file, "%d", &n);

mas = (char\*)malloc(n \* sizeof(char));

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

{

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

}

scanf\_s("%d", &a);

if (a == 1)

{

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

{

printf\_s("mas[%d] = %d\n", i, mas[i]);

}

}

if (a == 2)

{

printf\_s("Команды:\n1 - Пузырёк\n2 - Вставкой\n3 - Быстрая\n");

scanf\_s("%d", &a);

if (a == 1)

{

start = clock();

BubbleSort(mas, n);

end = clock();

tim = (double)(end - start) / CLOCKS\_PER\_SEC;

printf\_s("Время работы: %lf с\n", tim);

}

if (a == 2)

{

start = clock();

Vstavkasort(mas, n);

end = clock();

tim = (double)(end - start) / CLOCKS\_PER\_SEC;

printf\_s("Время работы: %lf с\n", tim);

}

if (a == 3)

{

start = clock();

Quicksort(mas, n);

end = clock();

tim = (double)(end - start) / CLOCKS\_PER\_SEC;

printf\_s("Время работы: %lf с\n", tim);

}

}

if (a == 3)

{

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

{

mas[i] = 0;

}

}

if (a == 4)

{

break;

}

**4.2. Описание алгоритмов*.***

Представим алгоритмы, которые используются в программе с помощью блок схем и фрагментов кода.

**4.2.1. Алгоритм считывания чисел из файла.**

char\* mas = NULL;

FILE\* file = 0;

file = fopen("input.txt", "r");

fscanf\_s(file, "%d", &n);

mas = (char\*)malloc(n \* sizeof(char));

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

{

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

}

Фрагмент кода 1. Считывание чисел

**4.2.2. Алгоритм консольного интерфейса (**См. Рисунок 10)

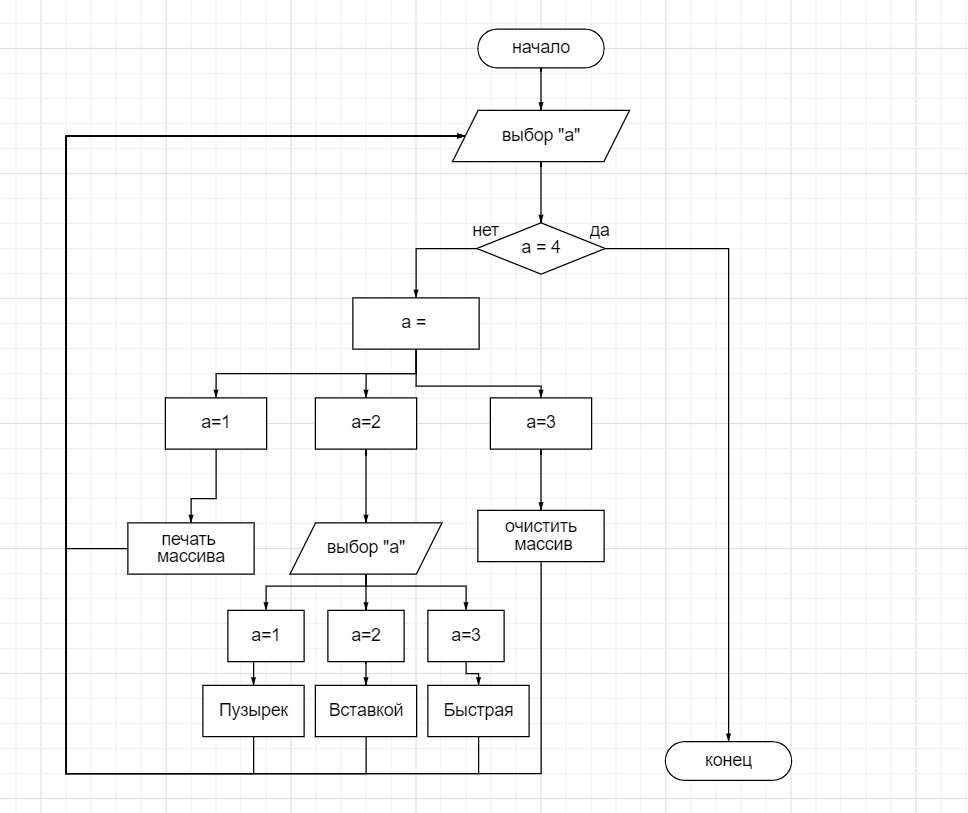


Рис.10. Блок схема. Консольный интерфейс

**4.2.3. Сортировка пузырьком.**

void BubbleSort(char\* a, int l)

{

int i, j;

int tmp = 0;

for (i = 0; i < l; i++)

{

for (j = 0; j < l - i - 1; j++)

{

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

{

tmp = a[j];

a[j] = a[j + 1];

a[j + 1] = tmp;

}

}

}

}

Фрагмент кода 1. Сортировка пузырьком

**4.2.4. Сортировка вставкой.**

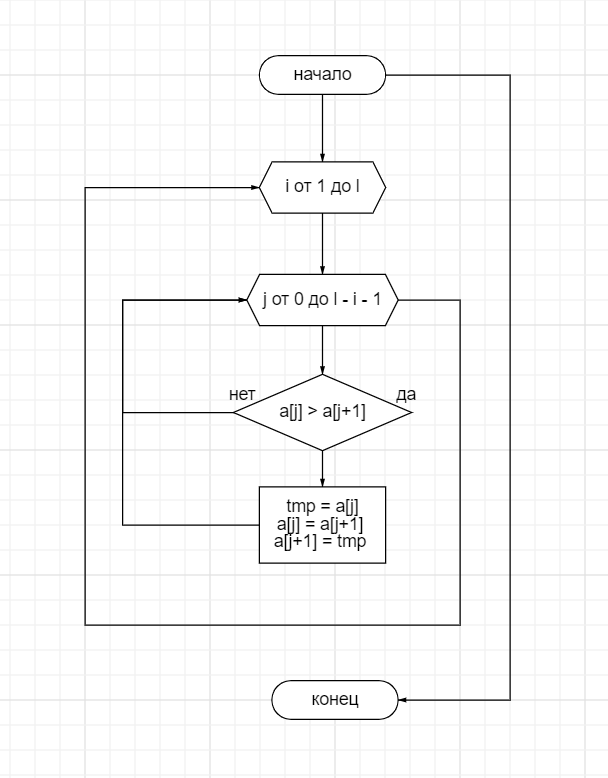
****

Рис.11. Блок схема.. Сортировка вставкой

void Vstavkasort(char\* a, int l)

{

int i, j;

int tmp = 0;

for (i = 1; i < l; i++)

{

tmp = a[i];

for (j = i - 1; j >= 0; j--)

{

if (a[j] > tmp)

{

a[j + 1] = a[j];

}

else

break;

}

a[j + 1] = tmp;

}

}

Фрагмент кода 2. Сортировка вставкой

**4.2.5. Быстрая сортировка.**

void Qsort(char\* mas, int l, int u)

{

int i = l;

int j = u;

int tmp = 0;

int x = mas[(int)((l + u) / 2)];

do

{

while (mas[i] < x)

++i;

while (mas[j] > x)

--j;

if (i <= j)

{

tmp = mas[i];

mas[i] = mas[j];

mas[j] = tmp;

i++;

j--;

}

} while (i < j);

if (l < j)

Qsort(mas, l, j);

if (i < u)

Qsort(mas, i, u);

}

void Quicksort(char\* a, int l)

{

Qsort(a, 0, (l - 1));

}

Фрагмент кода 3. Быстрая сортировка

**5. Эксперименты**

Используем таблицу для наглядного рассмотрения результатов. Введём разное количество данных и сравним время работы.

|  |  |  |
| --- | --- | --- |
| **Тип сортировки** | **Размер входных данных** | **Время работы, с** |
| Пузырьком | 10000 |  |
| Вставками | 10000 |  |
| Быстрая | 10000 |  |
| Пузырьком | 100000 |  |
| Вставками | 100000 |  |
| Быстрая | 100000 |  |

# **6.Заключение.**

В ходе лабораторной работы была написана программа на языке программирования «С», которая полностью выполняет поставленную задачу, а именно:

«Сравнение сортировок.

Реализовать сортировки массивов данных (тип данных “double”) задаваемых: обязательно случайно, дополнительно с клавиатуры или из файла.

Реализовать сортировки: пузырьком, вставкой, быстрая.

Сравнить время работы, сделать выводы.

Первая программа создает текстовый файл с записанными в него числами. Программа принимает количество чисел n, максимальное и минимальное значение.

Вторая программа читает текстовый файл с набором чисел, выводит консольный интерфейс (печать, сортировка, сброс, выход), выполняет выбранные действия».

Интерфейс программы простой, понятный и удобный в использовании, все сообщения выводятся на русском языке и при своих небольших размерах содержат нужную информацию, которую необходимо донести до пользователя.

В ходе сравнения сортировок можно сделать вывод: «быстрая сортировка (quick sort)» выполняет сортировку чисел быстрее остальных типов сортировки; «сортировка вставками» работает медленнее всех на большом диапазоне чисел, ее следует использовать на коротком отрезке не больше, чем десятки, или уже на частично отсортированном массиве данных.

# **Литература**

1. Бьерн Страуструп. Язык программирования С++ – Режим доступа: <http://8361.ru/6sem/books/Straustrup-Yazyk_programmirovaniya_c.pdf>
2. Т.А. Павловская Учебник по программированию на языках высокого уровня(С/С++) – Режим доступа: <http://cph.phys.spbu.ru/documents/First/books/7.pdf>

**Приложение 1**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

#include<stdlib.h>

#include<locale.h>

#include<time.h>

int main()

{

setlocale(LC\_ALL, "Rus");

int n = 0;

int min=0, max=0 ;

char\* mas = NULL;

FILE\* file;

srand(time(NULL));

file = fopen("C:/Users/Катя/source/repos/laba2.2/laba2.2/input.txt", "w");

printf\_s("Введите количество чисел n = ");

scanf\_s("%d", &n);

fprintf(file, "%d\n", n);

printf\_s("Введите значение (от -128 до 127) min = ");

scanf\_s("%d", &min);

fprintf(file, "%d\n", min);

printf\_s("Введите значение (от -128 до 127) max = ");

scanf\_s("%d", &max);

fprintf(file, "%d\n", max);

if ((max < min) || (n <= 0) )

{

printf("Ошибка: некорректные данные\n");

return 0;

}

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

{

char a = min + rand() % (max - min + 1);

fprintf(file, "%d\n", a);

}

return 0;

}

**Приложение 2**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

#include <stdlib.h>

#include <math.h>

#include <locale.h>

#include <time.h>

void BubbleSort(char\* a, int l)

{

int i, j;

int tmp = 0;

for (i = 0; i < l; i++)

{

for (j = 0; j < l - i - 1; j++)

{

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

{

tmp = a[j];

a[j] = a[j + 1];

a[j + 1] = tmp;

}

}

}

}

void Vstavkasort(char\* a, int l)

{

int i, j;

int tmp = 0;

for (i = 1; i < l; i++)

{

tmp = a[i];

for (j = i - 1; j >= 0; j--)

{

if (a[j] > tmp)

{

a[j + 1] = a[j];

}

else

break;

}

a[j + 1] = tmp;

}

}

void Qsort(char\* mas, int l, int u)

{

int i = l;

int j = u;

int tmp = 0;

int x = mas[(int)((l + u) / 2)];

do

{

while (mas[i] < x)

++i;

while (mas[j] > x)

--j;

if (i <= j)

{

tmp = mas[i];

mas[i] = mas[j];

mas[j] = tmp;

i++;

j--;

}

} while (i < j);

if (l < j)

Qsort(mas, l, j);

if (i < u)

Qsort(mas, i, u);

}

void Quicksort(char\* a, int l)

{

Qsort(a, 0, (l - 1));

}

int main()

{

setlocale(LC\_ALL, "Rus");

int n = 0, a = 0;

clock\_t start,end;

double tim = 0.0;

printf\_s("Команды:\n1 - Печать массива\n2 - Сортировка\n3 - Очистить массив\n4 - Завершить работу\n");

while (a != 4)

{

char\* mas = NULL;

FILE\* file = 0;

file = fopen("input.txt", "r");

fscanf\_s(file, "%d", &n);

mas = (char\*)malloc(n \* sizeof(char));

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

{

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

}

scanf\_s("%d", &a);

if (a == 1)

{

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

{

printf\_s("mas[%d] = %d\n", i, mas[i]);

}

}

if (a == 2)

{

printf\_s("Команды:\n1 - Пузырёк\n2 - Вставкой\n3 - Быстрая\n");

scanf\_s("%d", &a);

if (a == 1)

{

start = clock();

BubbleSort(mas, n);

end = clock();

tim = (double)(end - start) / CLOCKS\_PER\_SEC;

printf\_s("Время работы: %lf с\n", tim);

}

if (a == 2)

{

start = clock();

Vstavkasort(mas, n);

end = clock();

tim = (double)(end - start) / CLOCKS\_PER\_SEC;

printf\_s("Время работы: %lf с\n", tim);

}

if (a == 3)

{

start = clock();

Quicksort(mas, n);

end = clock();

tim = (double)(end - start) / CLOCKS\_PER\_SEC;

printf\_s("Время работы: %lf с\n", tim);

}

}

if (a == 3)

{

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

{

mas[i] = 0;

}

}

if (a == 4)

{

break;

}

printf\_s("Введите число\n");

}

return 0;

}