Национальный исследовательский

Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

Отчёт по лабораторной работе

**Сравнение сортировок.**

Выполнил:

студент ф-та ИИТММ гр. 381906-2

Сорокина Е.В.

Проверил:

преподаватель каф. МОСТ, ИИТММ

Лебедев И.Г.

Нижний Новгород

2019 г.

Содержание

[Введение 3](#_Toc26877535)

[Постановка задачи 4](#_Toc26877536)

[Руководство пользователя 5](#_Toc26877537)

[Руководство программиста 6](#_Toc26877538)

[Описание структуры программы 6](#_Toc26877539)

[Описание структур данных 7](#_Toc26877540)

[Описание алгоритмов 7](#_Toc26877541)

[Эксперименты 8](#_Toc26877542)

[Заключение 11](#_Toc26877543)

[Литература 12](#_Toc26877544)

[Приложения 13](#_Toc26877545)

[Приложение 1(заголовочный файл) 13](#_Toc26877546)

[Приложение 2 13](#_Toc26877547)

[Приложение 3 14](#_Toc26877548)

# Введение

Сортировка — один из базовых видов активности или действий, выполняемых над предметами. Ещё в детстве детей учат сортировать, развивая мышление. Компьютеры и программы — тоже не исключение. Цель сортировки - облегчить последующий поиск элементов в отсортированном множестве. Поэтому элементы сортировки присутствуют во многих задачах прикладного программирования.

В данной лабораторной работе мы рассмотрим сортировки нескольких видов: пузырьком (Bubble Sort), вставками (Insertion Sort) и быструю (Quick Sort) – и потом сравним их.

# Постановка задачи

Сравнение сортировок.

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.

# Руководство пользователя

Данная программа предполагается для использования, в первую очередь, в рамках основной задачи, описанной в предыдущем разделе отчёта. Однако помимо этого данную программу можно использовать также для создания и выведения на экран массива из n элементов в заданном диапазоне(от min до max включительно), записи в файл и считывания из него же созданного массива и сортировки его одним из выбранных способов, подсчёта времени, затраченного на сортировку каждого вида. Руководство программиста

## Описание структуры программы

Данная программа состоит из 2 проектов, каждый из которых включает несколько модулей. Код программы написан на языке C, что исключает использование классов. В первом проекте присутствуют 2 модуля: void chisla\_massiva(int n,short int min,short int max, short int \*\*mas), создающего массив размера n и заполняющего его случайными числами в диапазоне от min до max, и int main1(), который создаёт текстовый файл, записывает туда числа массива и его размер, а потом заново считывает их из него и выводит на экран. Во втором проекте присутствуют 4 модуля, первые три из которых ( void vstavkami(int n,short int\*\* mas) , void hoara(int l, int r, short int\*\* mas) и void puzirkom(int n, short int\*\* mas) ) реализуют сортировку массива вставками, быструю или пузырьком. Последний модуль во втором проекте (int main(short int\* mas, short int\* mas\_copy) ) является главным модулем во всей программе, и именно из него происходит вызов всех остальных функций(модулей). Также главный модуль производит замер временного интервала, за который массив сортируется указанным способом, и реализует вызов консоли, предоставляющей выбор для программиста: какое из 4 действий он хочет выбрать(печать, сортировка, сброс или выход).

Алгоритм работы программы построен следующим образом: сначала идёт ввод запрашиваемых данных (n, min, max) и проверка их достоверности (например, из соображений логики n не может иметь отрицательных значений); далее создаётся массив, который заполняется случайными числами в диапазоне от min до max, после чего происходит создание текстового файла, куда записываются числа массива и его размер, и считывание из него с последующим выводом на консоль записанного массива; после этого выводится консоль с выбором из 4 опций (печать, сортировка, сброс, выход), в зависимости от выбора одной из которых происходит выбранное действие. При выборе «печать» массив выводится на экран, при выборе «сортировка» консоль запрашивает вид сортировки (вставками, хоара или пузырьком), который вы хотите осуществить, и производит его с выведением подсчитанного времени, которое было затрачено данным алгоритмом, при выборе «сброс» отсортированный массив снова возвращается к своему начальному, неотсортированному состоянию и, наконец, при выборе «выход» программа освобождает память массива и завершает свою работу.

## Описание структур данных

Код данной программы, во-первых, как написанный на языке C, не может содержать классы, а, во-вторых, написан таким образом, чтобы обойтись без использования структур и глобальных переменных.

## Описание алгоритмов

В данной программе применяются такие алгоритмы, как

1. алгоритмы сортировки:



Рисунок 1. Сортировка пузырьком



Рисунок 2. Сортировка "Хоара"(быстрая)



Рисунок 3. Сортировка вставками

1. алгоритмы, связанные с массивом (создание и заполнение случайными числами, вывод на экран).

# Эксперименты

Эксперимент 1 : введение отрицательного количества элементов, n<0.

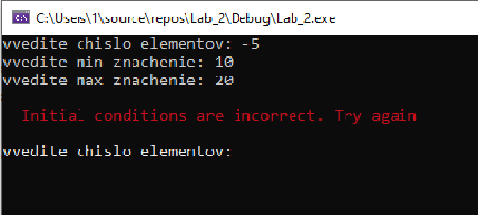


Рисунок 4. Эксперимент 1 - отрицательное кол-во элементов

Эксперимент 2 : введение неверного диапазона чисел, min>max .

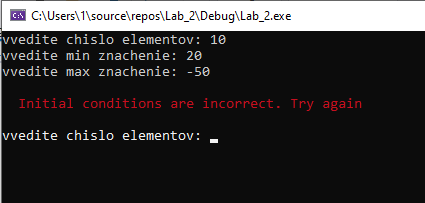


Рисунок 5. Эксперимент 2 - неправильный диапазон чисел

Эксперимент 3 : введение одинаковых значений min и max.

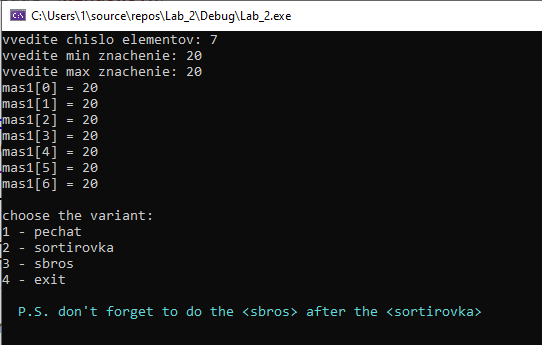
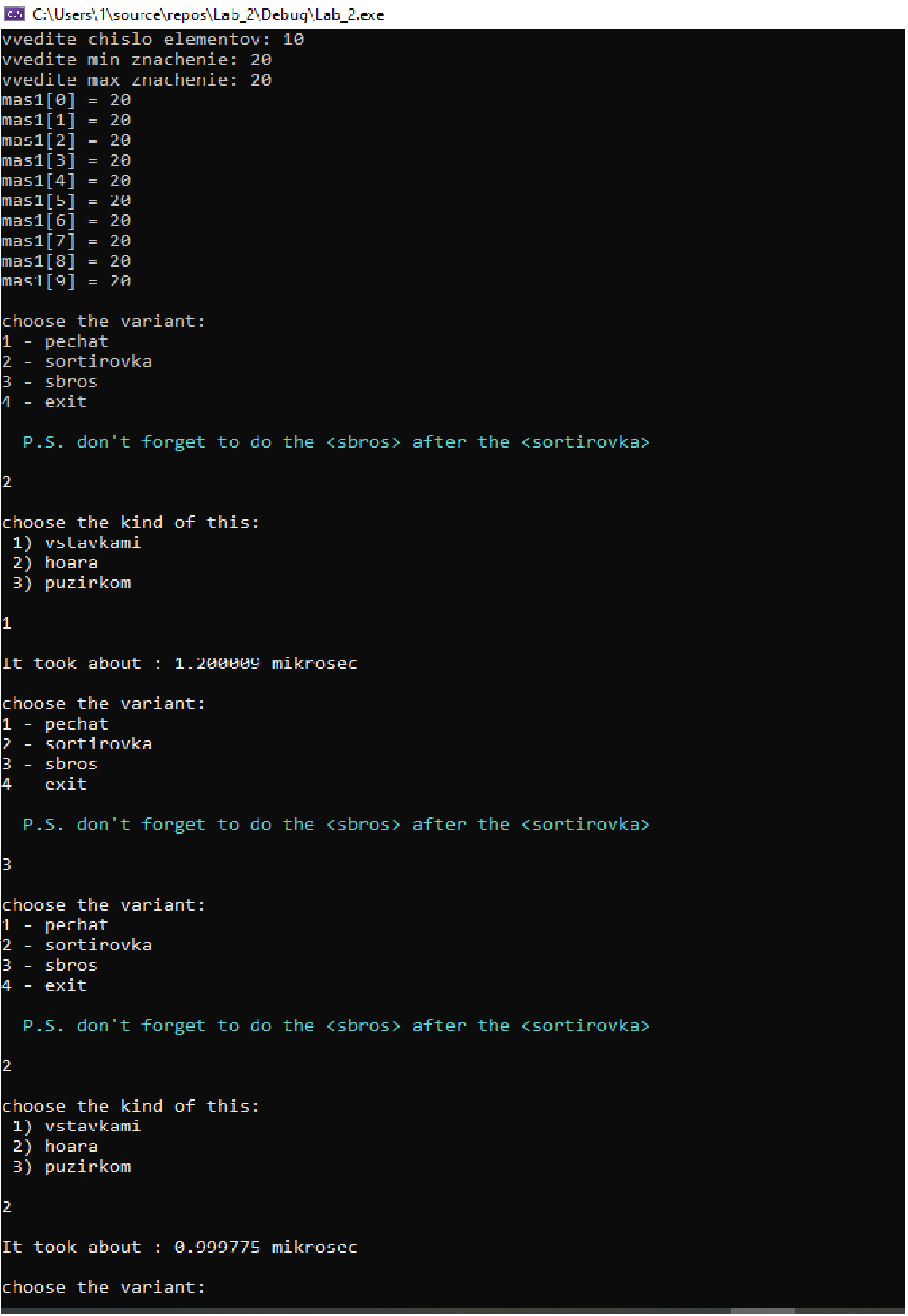


Рисунок 6. Эксперимент 3 - одинаковые значения min и max

Эксперимент 4 : проверка времени, которое уйдёт на сортировку массива, все элементы которого имеют одинаковые значения.



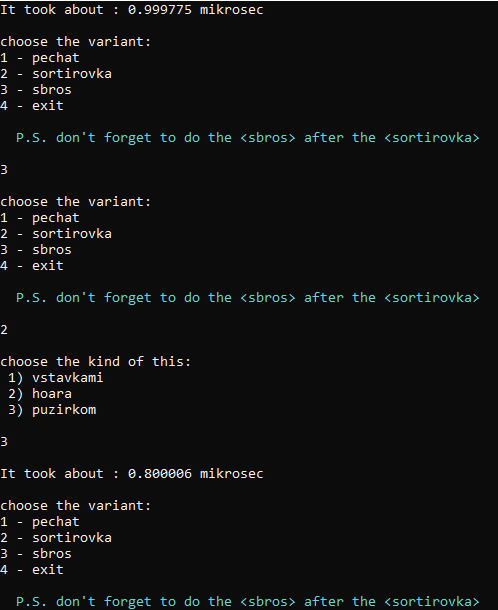


Рисунок 7. Эксперимент 4 - время сортировок массива из одинаковых элементов

Эксперимент 5 : работа программы при n=0.

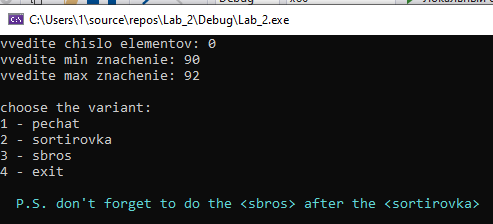


Рисунок 8. Эксперимент 5 - работа программы при n=0

# Заключение

Нам удалось достичь таких результатов, как создания и выведения на экран массива из n элементов в заданном диапазоне(от min до max включительно), записи в файл и считывания из него же созданного массива и сортировки его одним из выбранных способов. И, наконец, самое главное – нам удалось написать код, выполняющий поставленную задачу: наглядная демонстрация времени, затраченного каждой сортировкой, для удобного сравнения.

# Литература

1. Работа с текстовыми файлами - [https://learnc.info/c/text\_files/html]
2. Урок №66. Оператор goto – [https://ravesli.com/urok-66-operator-goto/]
3. Сортировка вставками(INSERTION-SORT) - [https://forkettle.ru/vidioteka/programmirovanie-i-set/108-algoritmy-i-struktury-dannykh/sortirovka-i-poisk-dlya-chajnikov/1005-sortirovka-vstavkami-insertion-sort]
4. 3.4.3. Функции для работы с системным таймером – [http://hpc.sfedu.ru/tutor/high\_performance\_computing/chapter3/page19.html]
5. Сортировка массива методом пузырька – [https://code-live.ru/solutions/cpp/3/]
6. Метод Хоара - Быстрая сортировка(Quick-sort) – [https://forkettle.ru/vidioteka/programmirovanie-i-set/108-algoritmy-i-struktury-dannykh/sortirovka-i-poisk-dlya-chajnikov/1010-metod-khoara-bystraya-sortirovka-quick-sort]
7. Как скомпилировать проект C с более чем одной основной функцией? – [https://qarchive.ru/644049\_kak\_skompilirovat\_\_proekt\_c\_s\_bolee\_chem\_odnoi\_\_osnovnoi\_\_funktsiei\_\_]
8. Секреты программирования. Изменение цвета в консоли (Си, Си++) – [https://mvblog.ru/archives/144/]

# Приложения

## Приложение 1(заголовочный файл)

#include <stdio.h>

#include <stdlib.h>

void chisla\_massiva(int n, short int min, short int max, short int\*\* mas);

int main1();

void vstavkami(int n, short int\*\* mas);

void hoara(int l, int r, short int\*\* mas);

void puzirkom(int n, short int\*\* mas);

## Приложение 2

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <Windows.h>

void chisla\_massiva(int n,short int min,short int max, short int \*\*mas)

{

int i = 0;

(\*mas) = (short int\*)malloc(n\*sizeof(short int));

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

{

(\*mas)[i] = min + rand()%(max - min + 1) ;

}

}

int main1()

{

FILE\* file;

int i = 0,j=0,n;

short int min, max;

short int\* mas;

short int\* mas1;

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

try\_again:

printf("vvedite chislo elementov: ");

scanf("%d",&n);

printf("vvedite min znachenie: ");

scanf("%hd", &min);

printf("vvedite max znachenie: ");

scanf("%hd", &max);

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

{

SetConsoleTextAttribute(hConsole, (WORD)((0 << 4) | 4));

printf("\n Initial conditions are incorrect. Try again\n\n");

SetConsoleTextAttribute(hConsole, (WORD)((0 << 4) | 15));

goto try\_again;

}

else

{

file = fopen("Prog1.txt","w");

if (file == NULL)

{

printf("error\n");

return 1;

}

chisla\_massiva(n,min,max,&mas);

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

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

{

fprintf(file,"mas[%d] = %hd\n",i,mas[i]);

}

fclose(file);

free(mas);

file = fopen("Prog1.txt", "r");

if (file == NULL)

{

printf("error\n");

return 1;

}

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

mas1 = (short int\*)malloc(n \* sizeof(short int));

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

{

fscanf(file,"mas[%d] = %hd\n",&j,&mas1[j]);

printf("mas1[%d] = %hd\n", j, mas1[j]);

j++;

}

fclose(file);

free(mas1);

return 0;

}

}

## Приложение 3

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "Header.h"

#include <stdio.h>

#include <stdlib.h>

#include <omp.h>

#include <Windows.h>

void vstavkami(int n,short int\*\* mas)

{

int i = 0, j = 0;

short int t;

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

{

t = (\*mas)[i];

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

{

if ((\*mas)[j] < t)

break;

(\*mas)[j + 1] = (\*mas)[j];

(\*mas)[j] = t;

}

}

}

void hoara(int l, int r, short int\*\* mas)

{

int i = l, j = r;

short int t,x = (\*mas)[(l + r) / 2];

do

{

while ((\*mas)[i] < x)

i++;

while ((\*mas)[j] > x)

j--;

if (i <= j)

{

if ((\*mas)[i] > (\*mas)[j])

{

t = (\*mas)[i];

(\*mas)[i] = (\*mas)[j];

(\*mas)[j] = t;

}

i++;

j--;

}

}

while (i <= j);

if (i < r)

hoara( i, r,mas);

if (l < j)

hoara( l, j,mas);

}

void puzirkom(int n, short int\*\* mas)

{

int i = 0, j = 0;

short int t;

for (int i = 0; i < n - 1; i++)

{

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

{

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

{

t = (\*mas)[j];

(\*mas)[j] = (\*mas)[j + 1];

(\*mas)[j + 1] = t;

}

}

}

}

int main(short int\* mas, short int\* mas\_copy)

{

FILE\* File;

int i = 0,j=0, n;

int var, var1;

long double start\_time =0, end\_time=0;

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

main1();

File = fopen("Prog1.txt","r");

fscanf(File, "%d\n", &n);

mas = (short int\*)malloc(n\*sizeof(short int));

mas\_copy = (short int\*)malloc(n \* sizeof(short int));

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

{

fscanf(File, "mas[%d] = %hd\n", &j, &mas[j]);

mas\_copy[j] = mas[j];

j++;

}

fclose(File);

again:

printf("\nchoose the variant:\n");

printf("1 - pechat\n");

printf("2 - sortirovka\n");

printf("3 - sbros\n");

printf("4 - exit\n");

SetConsoleTextAttribute(hConsole, (WORD)((0 << 4) | 11));

printf("\n P.S. don't forget to do the <sbros> after the <sortirovka>\n\n");

SetConsoleTextAttribute(hConsole, (WORD)((0 << 4) | 15));

scanf("%d", &var);

switch (var)

{

case(1):

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

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

goto again;

case(2):

printf("\nchoose the kind of this:\n");

printf(" 1) vstavkami\n");

printf(" 2) hoara\n");

printf(" 3) puzirkom\n\n");

scanf("%d", &var1);

switch (var1)

{

case(1):

start\_time = omp\_get\_wtime();

vstavkami(n, &mas);

end\_time = omp\_get\_wtime();

printf("\nIt took about : %0.6lf mikrosec\n", (end\_time - start\_time) \* 1000000);

start\_time = 0;

end\_time = 0;

goto again;

case(2):

start\_time = omp\_get\_wtime();

hoara(0,n-1,&mas);

end\_time = omp\_get\_wtime();

printf("\nIt took about : %0.6lf mikrosec\n", (end\_time - start\_time) \* 1000000);

start\_time = 0;

end\_time = 0;

goto again;

case(3):

start\_time = omp\_get\_wtime();

puzirkom(n, &mas);

end\_time = omp\_get\_wtime();

printf("\nIt took about : %0.6lf mikrosec\n",(end\_time - start\_time)\*1000000);

start\_time = 0;

end\_time = 0;

goto again;

}

case(3):

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

mas[i] = mas\_copy[i];

goto again;

case(4):

free(mas);

free(mas\_copy);

break;

}

}