Министерство образования и науки Российской Федерации

Федеральное агентство по образованию

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Лабораторная работа №5

по курсу «Программирование»

Исследование алгоритмов сортировки

Выполнил студент группы ИВТ-11 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Птахова А.М/

Проверил преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Чистяков Г. А./

Киров 2021

**Цель работы**: получить базовые сведения о наиболее известных алгоритмах сортировки, изучить принципы работы с текстовыми файлами.

**Задание:**

1. Реализовать сортировку данных с помощью "пузырькового" алгоритма.

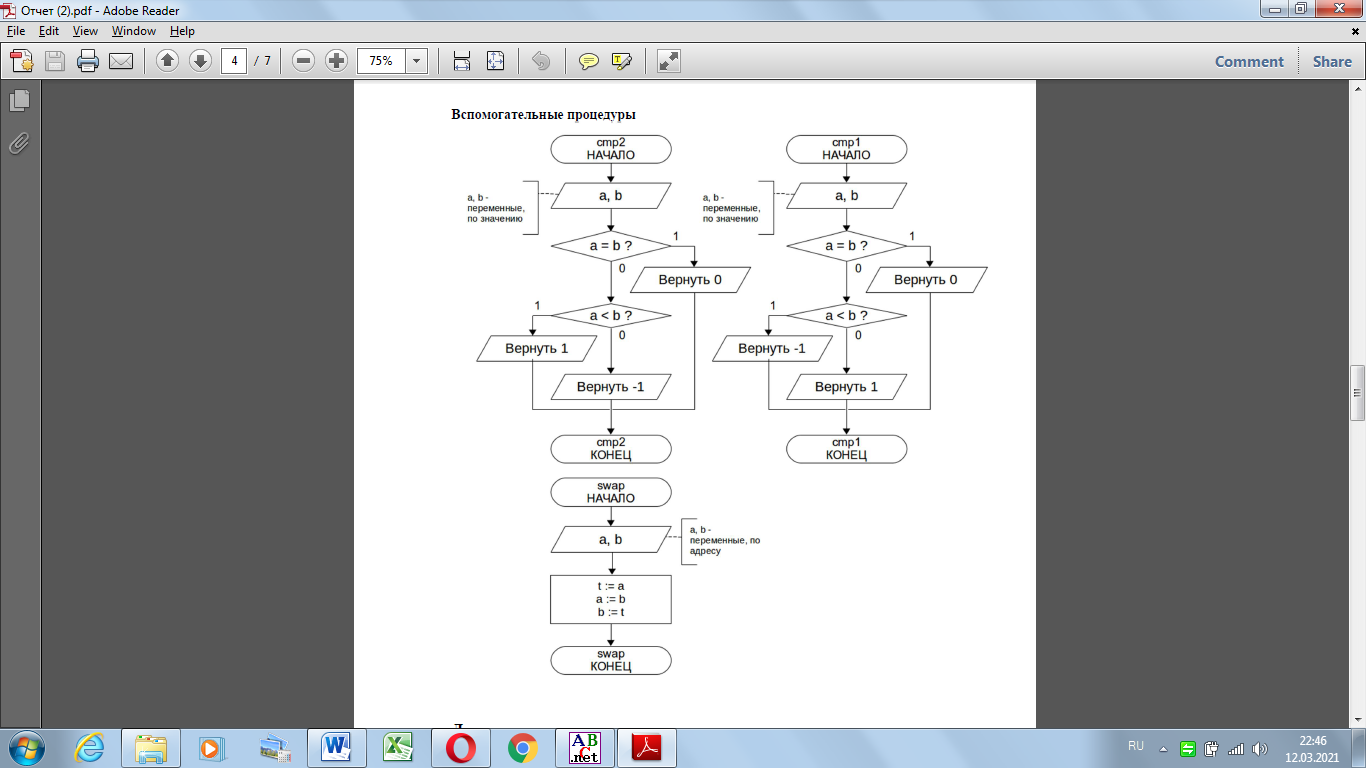
2. Реализовать сортировку данных с помощью быстрого алгоритма.

3. В обоих случаях необходимо предусмотреть возможность изменения компаратора (реализация компаратора в виде передаваемой в подпрограмму функции).

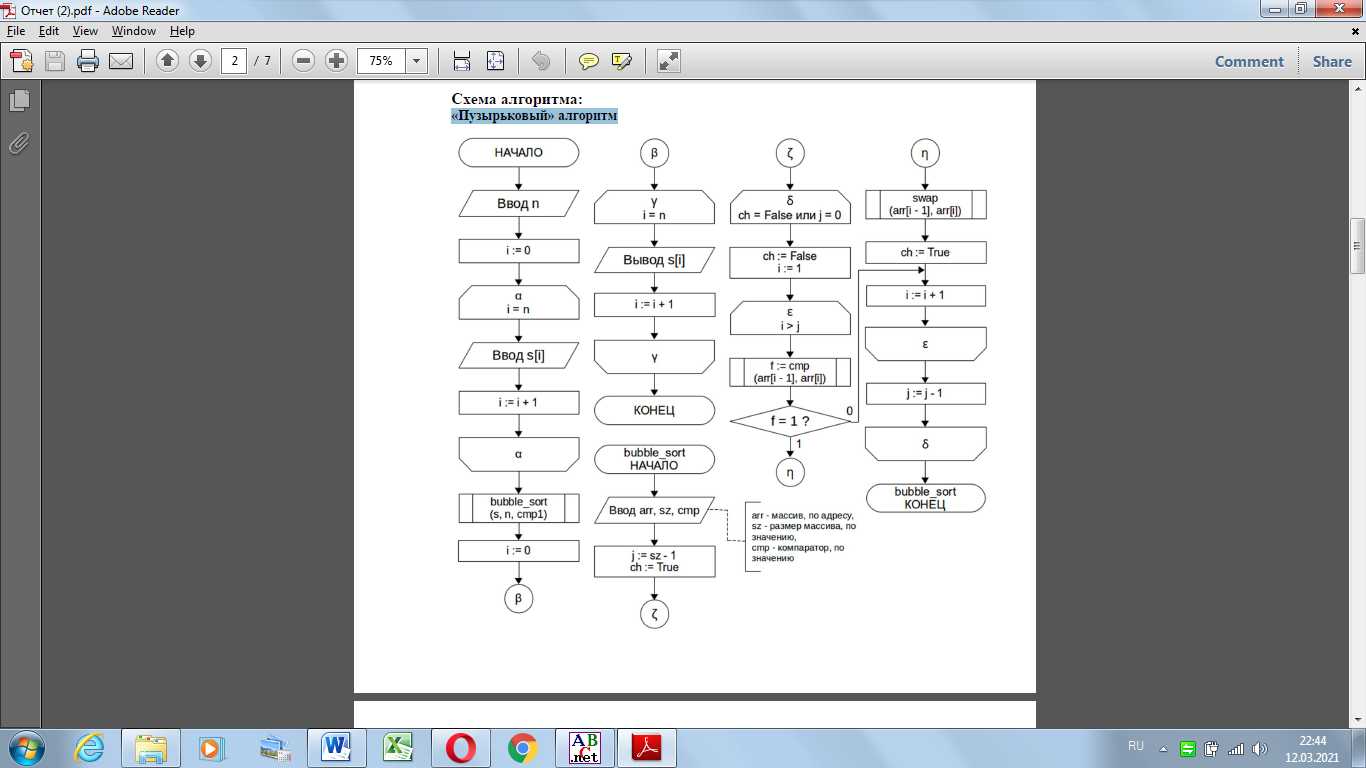
4. Считывание и вывод данных необходимо производить из текстового файла.

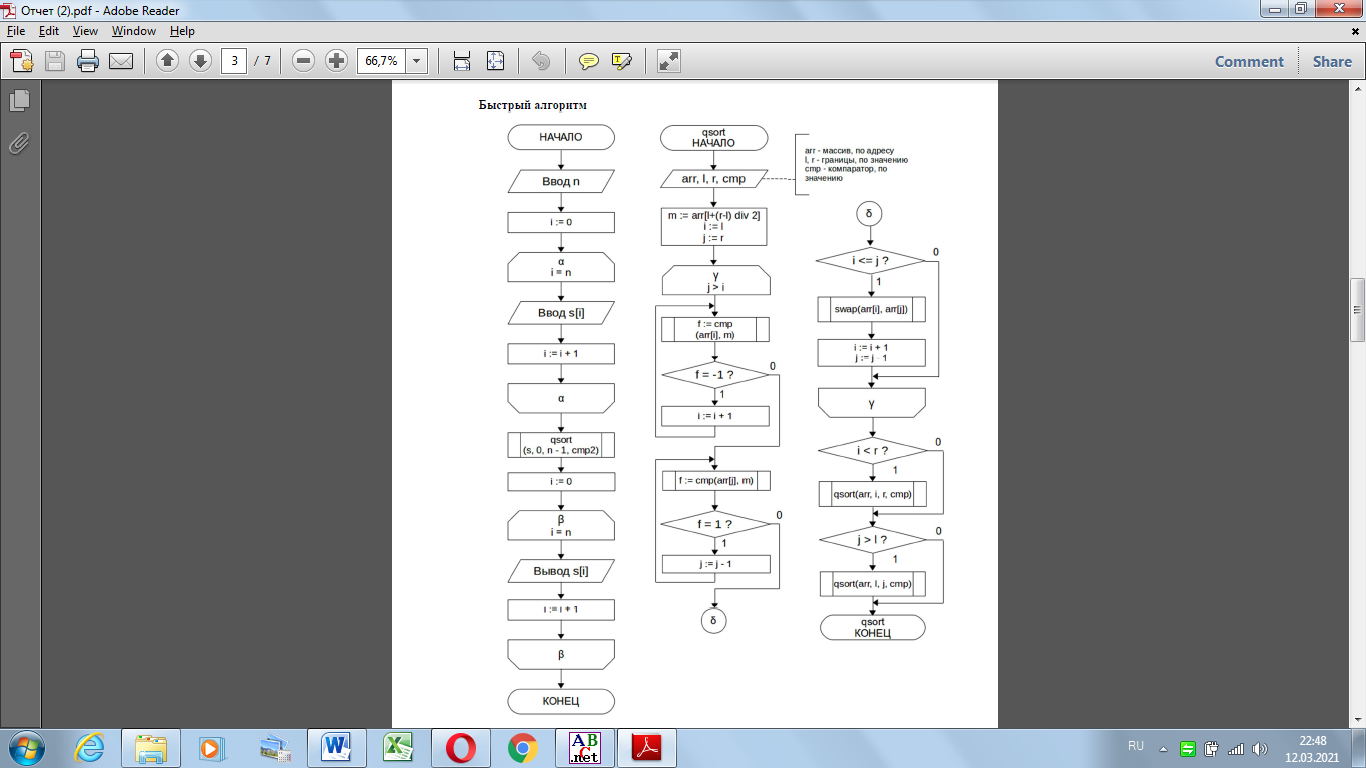
5. Для демонстрации работы программных реализаций самостоятельно подготовить варианты входных данных (при этом объем тестовых файлов должен позволять оценить скорость работы программ).

**Схема алгоритма:**



«Пузырьковый» алгоритм





**Листинг кода**:

«Пузырьковый» алгоритм

type cmp\_t = function(a, b : Longint):Integer;

array\_t = array [0..1000100] of Longint;

function cmp1(a, b : Longint):Integer;

begin

if a = b then cmp1 := 0

else if a < b then cmp1 := -1

else if a > b then cmp1 := 1;

end;

procedure swap(var a, b : Longint);

var t : Longint;

begin

t := a;

a := b;

b := t;

end;

procedure bubble\_sort(var arr : array\_t; sz : Longint; cmp : cmp\_t);

var ch : Boolean;

i, j : Longint;

begin

j := sz - 1;

ch := True;

while ch or (j > 0) do begin

ch := False;

for i := 1 to j do

if cmp(arr[i - 1], arr[i]) = 1 then begin

swap(arr[i - 1], arr[i]);

ch := True;

end;

dec(j);

end;

end;

var inp, out : TextFile;

s : array\_t;

n, i : Longint;

begin

assign(inp, 'input.txt');

assign(out, 'output.txt');

reset(inp);

rewrite(out);

readln(inp, n);

for i := 0 to n - 1 do read(inp, s[i]);

bubble\_sort(s, n, @cmp1);

for i := 0 to n - 1 do write(out, s[i], ' ');

close(inp);

close(out);

end.

Быстрый алгоритм

type cmp\_t = function(a, b : Longint):Integer;

array\_t = array [0..1000100] of Longint;

function cmp2(a, b : Longint):Integer;

begin

if a = b then cmp2 := 0

else if a < b then cmp2 := 1

else if a > b then cmp2 := -1;

end;

procedure swap(var a, b : Longint);

var t : Longint;

begin

t := a;

a := b;

b := t;

end;

procedure qsort(var arr : array\_t; l, r: Longint; cmp : cmp\_t);

var i, j : Longint;

m : Longint;

begin

m := arr[l + (r - l) div 2];

i := l;

j := r;

while i <= j do begin

while cmp(arr[i], m) = -1 do inc(i);

while cmp(arr[j], m) = 1 do dec(j);

if i <= j then begin

swap(arr[i], arr[j]);

inc(i);

dec(j);

end;

end;

if i < r then qsort(arr, i, r, cmp);

if j > l then qsort(arr, l, j, cmp);

end;

var inp, out : TextFile;

s : array\_t;

n, i : Longint;

begin

assign(inp, 'input.txt');

assign(out, 'output.txt');

reset(inp);

rewrite(out);

readln(inp, n);

for i := 0 to n - 1 do read(inp, s[i]);

qsort(s, 0, n - 1, @cmp2);

for i := 0 to n - 1 do write(out, s[i], ' ');

close(inp);

close(out);

end.

**Вывод**: В данной лабораторной работе были получены базовые сведения о наиболее известных алгоритмах сортировки, изучены принципы работы с текстовыми файлами.