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-12 /Крючков И. С/ Проверил /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 алгоритма подсчетом.

2. Реализовать сортировку данных с помощью поразрядного 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:**

**Сортировка подсчетом по возрастанию**

**E:\Универ\2 семестр\Программирование\Лабы\5\Схемы\lab5.1.1.emf**

**Сортировка подсчетом по убыванию**

**E:\Универ\2 семестр\Программирование\Лабы\5\Схемы\lab5.1.2.emf**

**Поразрядная сортировка по возрастанию**

**E:\Универ\2 семестр\Программирование\Лабы\5\Схемы\lab5.2.1.1.emf**

**E:\Универ\2 семестр\Программирование\Лабы\5\Схемы\lab5.2.1.2.emf**

**Поразрядная сортировка по убыванию**

**E:\Универ\2 семестр\Программирование\Лабы\5\Схемы\lab5.2.2.1.emf**

**E:\Универ\2 семестр\Программирование\Лабы\5\Схемы\lab5.2.1.2.emf**

**Экранная форма:**





**Листинг кода:**

**Сортировка подсчетом по возрастанию**

program podschet\_v;

type

massivA = array[0..1000] of longint;

massivB = array[-10000..10000] of longint;

procedure sort(var a:massivA; b:massivB; n:longint);

var

i,j,idx:integer;

begin

for i := 0 to n-1 do

begin

inc(b[a[i]]);

end;

idx := 0;

for i:=-10000 to 10000 do

begin

j := 0;

while(j < b[i]) do

begin

a[idx] := i;

inc(idx);

inc(j);

end;

end;

end;

var

a:massivA;

b:massivB;

i:integer;

inp, out:text;

n:longint;

begin

assign(inp, 'input.txt');

assign(out, 'output.txt');

reset(inp);

rewrite(out);

readln(inp, n);

b[0] := 0;

for i := 0 to n - 1 do

read(inp, a[i]);

sort(a, b, n);

for i := 0 to n-1 do

write(out, a[i], ' ');

close(inp);

close(out);

end.

**Сортировка подсчетом по убыванию**

program podschet\_u;

type

massivA = array[0..1000] of longint;

massivB = array[-10000..10000] of longint;

procedure sort(var a:massivA; b:massivB; n:longint);

var

i,j,idx:integer;

begin

for i := 0 to n-1 do

begin

inc(b[a[i]]);

end;

idx := 0;

for i:=-10000 to 10000 do

begin

j := 0;

while(j < b[i]) do

begin

a[idx] := i;

inc(idx);

inc(j);

end;

end;

end;

var

a:massivA;

b:massivB;

i:integer;

inp, out:text;

n:longint;

begin

assign(inp, 'input.txt');

assign(out, 'output.txt');

reset(inp);

rewrite(out);

readln(inp, n);

b[0] := 0;

for i := 0 to n - 1 do

read(inp, a[i]);

sort(a, b, n);

for i := n-1 downto 0 do

write(out, a[i], ' ');

close(inp);

close(out);

end.

**Поразрядная сортировка по возрастанию**

program radix\_v;

type

massiv = array[1..1000000] of longint;

var

i,j,k,u,n,x,max,z,p: longint;

plus, minus: massiv;

inp, out: text;

procedure sort(var a:massiv; kol:longint);

var

i, mm:longint;

t, l:integer;

f:longint;

vs:massiv;

begin

for i := 1 to kol do

vs[i] := a[i];

f := 1;

j := 0;

mm := max;

if kol > 0 then

begin

while mm <> 0 do

begin

for l := 0 to 9 do

begin

for z := 1 to kol do

begin

t := vs[z] div f mod 10;

if t = l then

begin

inc(j);

a[j] := vs[z];

end;

end;

end;

for p := 1 to kol do

vs[p] := a[p];

j := 0;

f := f \* 10;

mm := mm div 10;

end;

end;

end;

begin

max := 0;

k := 0;

u := 0;

assign(inp, 'input.txt');

assign(out, 'output.txt');

reset(inp);

reWrite(out);

readln(inp,n);

for i := 1 to n do

begin

read(inp, x);

if abs(x) > max then

max := x;

if x < 0 then

begin

inc(k);

minus[k] := abs(x);

end

else

begin

inc(u);

plus[u] := x;

end;

end;

max := max \* 10;

sort(minus,k);

for i := k downto 1 do

write(out,-minus[i],' ');

sort(plus, u);

for i := 1 to u do

write(out, plus[i],' ');

close(inp);

close(out);

end.

**Поразрядная сортировка по убыванию**

program radix\_u;

type

massiv = array[1..1000000] of longint;

var

i,j,k,u,n,x,max,z,p:longint;

plus,minus:massiv;

inp, out:text;

procedure sort(var a:massiv; kol:longint);

var

i, mm:longint;

t, l:integer;

f:longint;

vs:massiv;

begin

for i := 1 to kol do

vs[i] := a[i];

f := 1;

j := 0;

mm := max;

if kol > 0 then

begin

while mm <> 0 do

begin

for l := 0 to 9 do

begin

for z := 1 to kol do

begin

t := vs[z] div f mod 10;

if t = l then

begin

inc(j);

a[j] := vs[z];

end;

end;

end;

for p := 1 to kol do

vs[p] := a[p];

j := 0;

f := f \* 10;

mm := mm div 10;

end;

end;

end;

begin

k := 0;

u := 0;

max := 0;

assign(inp, 'input.txt');

assign(out, 'output.txt');

reset(inp);

reWrite(out);

readln(inp,n);

for i := 1 to n do

begin

read(inp, x);

if abs(x) > max then

max := x;

if x < 0 then

begin

inc(k);

minus[k] := abs(x);

end

else

begin

inc(u);

plus[u] := x;

end;

end;

max := max \* 10;

sort(plus,u);

for i := u downto 1 do

write(out, plus[i], ' ');

sort(minus, k);

for i := 1 to k do

write(out, -minus[i], ' ');

close(inp);

close(out);

end.

**Вывод:** ходе выполнения лабораторной работы были получены базовые сведения о наиболее известных алгоритмах сортировки, изучены принципы работы с текстовыми файлами, а именно открытие текстовых файлов, закрытие, запись, чтение. Написаны сортировка подсчетом и поразрядная сортировка.