Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЫ №5**

**«ИСЛЕДОВАНИЕ АЛГОРИТМОВ СОРТИРОВКИ»**

**ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

Выполнил: студент учебной группы

ИСПк-204-52-00

Неганов Максим Дмитриевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2022

**Цель работы**

Получить базовые сведения о наиболее известных алгоритмах сортировки, изучить принципы работы с текстовыми файлами

**Формулировка задания**

Вариант 15

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 функций).

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

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

**Код программы**

**uses** crt;

**uses** System;

**var**

F\_in, F\_in1: **file of** char;

F\_out, F\_out1: text;

arr, arr1: **array** [0..100000] **of** char;

n, time\_in, time\_out, time1\_in, time1\_out: integer;

**function** puzvoz: integer;

**var**

c: char;

**begin**

reset(F\_in);

**for var** i := 1 **to** n **do**

read(F\_in, arr[i]);

**for var** i := 1 **to** n - 1 **do**

**begin**

**for var** j := n - 1 **downto** i **do**

**begin**

**if** arr[j] > arr[j + 1] **then**

**begin**

c := arr[j];

arr[j] := arr[j + 1];

arr[j + 1] := c;

**end**;

**end**;

**end**;

**for var** i := 1 **to** n **do**

write(arr[i], ' ');

**end**;

**function** puzubv: integer;

**var**

c: char;

**begin**

reset(F\_in);

**for var** i := 1 **to** n **do**

read(F\_in, arr[i]);

**for var** i := 1 **to** n - 1 **do**

**begin**

**for var** j := n - 1 **downto** i **do**

**begin**

**if** arr[j] < arr[j + 1] **then**

**begin**

c := arr[j];

arr[j] := arr[j + 1];

arr[j + 1] := c;

**end**;

**end**;

**end**;

**for var** i := 1 **to** n **do**

write(arr[i], ' ');

**end**;

**function** bstrvoz(first,last: integer):integer;

**var**

L, R: integer;

X,c:char;

**begin**

**if** first < last **then**

**begin**

X := arr1[(first + last) **div** 2];

L := first;

R := last;

**while** L <= R **do**

**begin**

**while** arr1[L] < X **do**

L := L + 1;

**while** arr1[R] > X **do**

R := R - 1;

**if** L <= R **then**

**begin**

c := arr1[L];

arr1[L] := arr1[R];

arr1[R] := c;

L := L + 1;

R := R - 1;

**end**;

**end**;

bstrvoz(first, R);

bstrvoz(L, last);

**end**;

**end**;

**function** bstrubv(last,first: integer):integer;

**var**

L, R: integer;

X,c:char;

**begin**

**if** first > last **then**

**begin**

X := arr1[(first + last) **div** 2];

L := first;

R := last;

**while** L <= R **do**

**begin**

**while** arr1[L] < X **do**

L := L + 1;

**while** arr1[R] > X **do**

R := R - 1;

**if** L <= R **then**

**begin**

c := arr1[L];

arr1[L] := arr1[R];

arr1[R] := c;

L := L + 1;

R := R - 1;

**end**;

**end**;

bstrubv(first, R);

bstrubv(L, last);

**end**;

**end**;

**function** sort(n:integer): integer;

**var**

a: integer;

**begin**

ClrScr;

writeln('Изменить сортировку');

writeln('Сортировка по возрастанию пузырьковым- 1');

writeln('Сортировка по убыванию пузырьковым- 2');

writeln('Сортировка по возрастанию быстрым- 3');

writeln('Сортировка по убыванию быстрым- 4');

writeln('Выход- 0');

readln(a);

**case** a **of**

1:

**begin**

time1\_in := DateTime.Now.Second;

time\_in := DateTime.Now.MilliSecond;

puzvoz;

time1\_out := DateTime.Now.Second;

time\_out := DateTime.Now.MilliSecond;

writeln('Время: ', time1\_out - time1\_in, ' секунд ', abs(time\_out - time\_in), ' милисекунд');

readln;

sort(n);

**end**;

2:

**begin**

time1\_in := DateTime.Now.Second;

time\_in := DateTime.Now.MilliSecond;

puzubv;

time1\_out := DateTime.Now.Second;

time\_out := DateTime.Now.MilliSecond;

writeln('Время: ', time1\_out - time1\_in, ' секунд ', abs(time\_out - time\_in), ' милисекунд');

readln;

sort(n);

**end**;

3:

**begin**

time1\_in := DateTime.Now.Second;

time\_in := DateTime.Now.MilliSecond;

reset(F\_in1);

**for var** i := 1 **to** n **do**

**begin**

read(F\_in1, arr1[i]);

**end**;

bstrvoz(1,n);

**for var** i := 1 **to** n **do**

**begin**

write(arr1[i], ' ');

**end**;

time1\_out := DateTime.Now.Second;

time\_out := DateTime.Now.MilliSecond;

writeln('Время: ', time1\_out - time1\_in, ' секунд ', abs(time\_out - time\_in), ' милисекунд');

readln;

sort(n);

**end**;

4:

**begin**

time1\_in := DateTime.Now.Second;

time\_in := DateTime.Now.MilliSecond;

reset(F\_in1);

**for var** i := 1 **to** n **do**

**begin**

read(F\_in1, arr1[i]);

**end**;

bstrubv(1,n);

**for var** i := 1 **to** n **do**

**begin**

write(arr1[i], ' ');

**end**;

time1\_out := DateTime.Now.Second;

time\_out := DateTime.Now.MilliSecond;

writeln('Время: ', time1\_out - time1\_in, ' секунд ', abs(time\_out - time\_in), ' милисекунд');

readln;

sort(n);

**end**;

0:

**end**;

**end**;

**function** vvod: integer;

**begin**

ClrScr;

writeln('Размер массива');

readln(n);

assign(F\_in, '1in.txt');

assign(F\_out, '1out.txt');

assign(F\_in1, '2in.txt');

assign(F\_out1, '2out.txt');

writeln('Содержимое 1-го массива: ');

rewrite(F\_in);

**var** a: char;

**for var** i := 1 **to** n **do**

**begin**

readln(a);

write(F\_in, a);

**end**;

rewrite(F\_in1);

writeln('Содержимое 2-го массива:');

**for var** i := 1 **to** n **do**

**begin**

readln(a);

write(F\_in1, a);

**end**;

sort(n);

rewrite(F\_out);

**for var** i:=1 **to** n **do**

**begin**

write(F\_out,arr[i]);

**end**;

rewrite(F\_out1);

**for var** i:=1 **to** n **do**

**begin**

write(F\_out1, arr1[i]);

**end**;

**end**;

**begin**

vvod;

close(F\_in1);

close(F\_out1);

close(F\_in);

close(F\_out);

**end**.

**Результат выполнения программы**

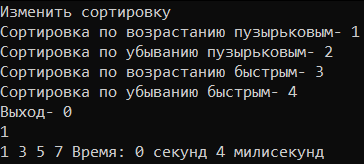


Рисунок 1 – Результат выполнения программы задачи (1 из 5)

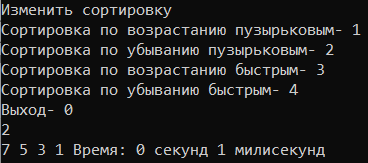


Рисунок 2 – Результат выполнения программы задачи (2 из 5)

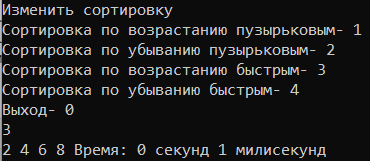


Рисунок 3 – Результат выполнения программы задачи (3 из 5)

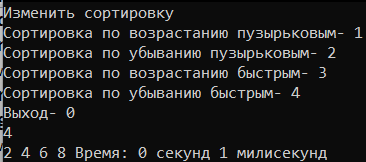


Рисунок 4 – Результат выполнения программы задачи (4/5)

**Вывод**

В данной программе мы рассмотрели несколько способов сортировки и научились применять их с данными, так же узнал о такой «библиотеке» как system, которая позволила вычислить время выполнения программы, так же вспомнили работу функций и case-меню.

В ходе выполнения работы трудностей не возникло.