Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №5**

**«ИССЛЕДОВАНИЕ АЛГОРИТМОВ СОРТИРОВКИ»**

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

Выполнил: студент учебной группы

ИСПк-203-52-00

Обухов Илья Николаевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

Содержание

1. Цель работы

2. Формулировка задания (с вариантом)

3. Описание алгоритмов

4. Схемы алгоритмов

5. Код программы

6. Результат выполнения программы

7. Вывод

**Цель работы:**

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

**Формулировка задания:**

Вариант: 17

1. Реализовать сортировку данных с помощью алгоритма подсчетом.
2. Реализовать сортировку данных с помощью слияния.
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 текстовых файлов должен позволять оценить скорость работы программ).

**3. Описание алгоритмов**

Сортировка **подсчетом** (sortObw):

Считывает данные из файла "D:\мал.txt" и подсчитывает количество вхождений каждого числа в массиве arr.

В зависимости от выбора пользователя (ch) сортирует и выводит отсортированный массив по возрастанию или убыванию.

Результат сохраняется в файл "D:\output.txt".

Сортировка **слиянием** (sortprost и sliynie):

sortprost - рекурсивно (обращение к самому себе) разбивает массив на подмассивы и вызывает процедуру sliynie для их слияния.

sortLoc - считывает данные из файла "D:\мал.txt" в массив a, в зависимости от выбора пользователя (xz) сортирует и выводит отсортированный массив по возрастанию или убыванию.

Результат сохраняется в файл "D:\output.txt".

Основная часть программы:

Использует консоль для ввода и вывода данных (uses crt).

Пользователю предоставляется выбор между сортировкой подсчетом и сортировкой слиянием.

Для каждого метода сортировки также предоставляется выбор направления сортировки.

Программа выполняется в цикле до тех пор, пока пользователь не выберет пункт "3. Выход".

Общее:

Используются файлы для ввода (D:\мал.txt) и вывода (D:\output.txt) данных.

Переменная xz используется для выбора метода сортировки и направления.

**4. Схема алгоритма**

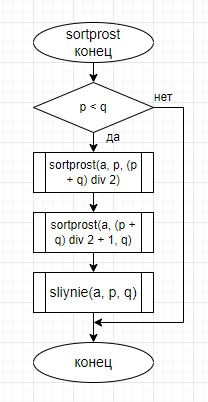


Рисунок 1 – Схема процедуры для сортировки частей массива в сортировке алгоритмом слияния.

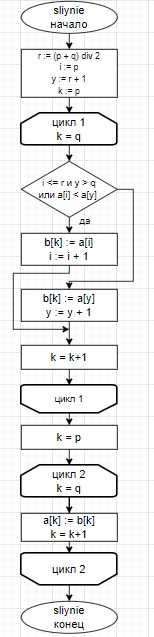


Рисунок 2 – Схема процедуры для слияния массива в один в сортировке алгоритмом слияния.

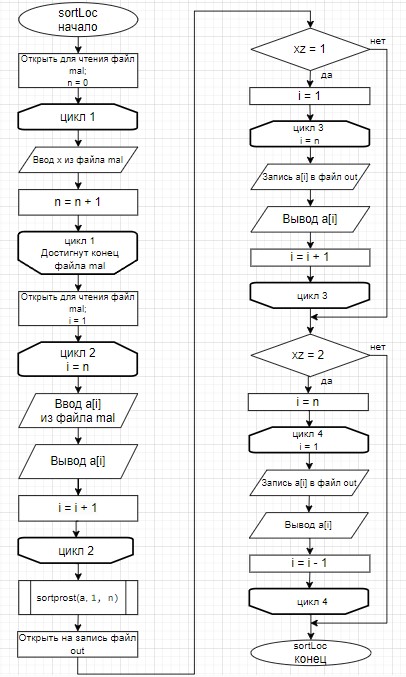


Рисунок 3 – Схема процедуры для сортировки алгоритмом слияния.

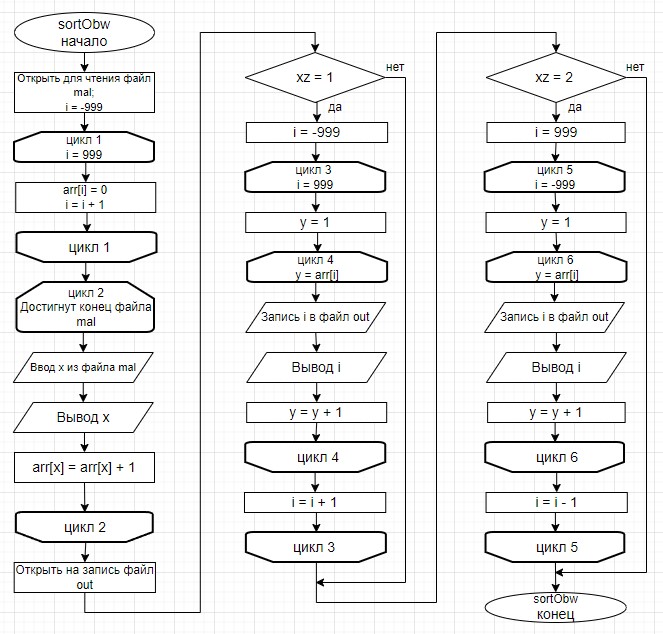


Рисунок 4 – Схема алгоритма для сортировки алгоритмом подсчета.

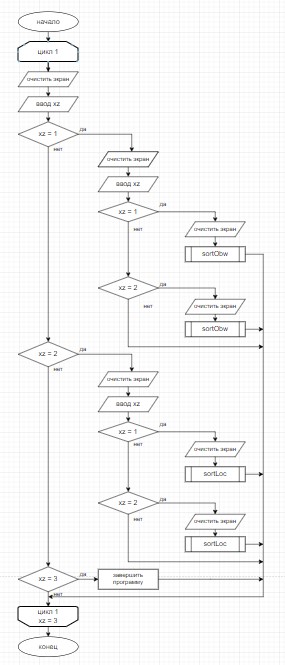


Рисунок 5 – Схема основной программы.

**5. Код программы:**

**program** dkr5;

**uses** crt; //консоль

**type**

ms = **array** [1..9999] **of** integer;

**var**

mal, out: text;

xz: integer;

**procedure** sortObw;

**var**

arr: **array**[-999..999] **of** integer;

i, y, x: integer;

**begin**

Assign(mal, 'D:\мал.txt'); //связывает файловую

//переменную с файлом на диске

Reset(mal); //открывает файл для чтения

Writeln('Наш массив: ');

**for** i := -999 **to** 999 **do**

arr[i] := 0;

**while not** eof(mal) **do** //возвращает true, когда конец файла

**begin**

readln(mal, x);

writeln(x, ' ');

inc(arr[x]);

**end**;

Assign(out, 'D:\output.txt');

ReWrite(out); //создает файл на запись

writeln;

writeln('Отсортированный массив: ');

**if** xz = 1 **then**

**begin**

**for** i := -999 **to** 999 **do**

**for** y := 1 **to** arr[i] **do**

**begin**

writeln(out, i);

write(i);

**end**;

**end**;

**if** xz = 2 **then**

**begin**

**for** i := 999 **downto** -999 **do**

**for** y := 1 **to** arr[i] **do**

**begin**

writeln(out, i);

write(i);

**end**;

**end**;

close(out); //закрывает файл

close(mal);

**end**;

**procedure** sliynie(**var** a: ms; p, q: integer);

**var**

r, i, y, k: integer;

b: ms;

**begin**

r := (p + q) **div** 2;

i := p;

y := r + 1;

**for** k := p **to** q **do**

**if** (i <= r) **and** ((y > q) **or** (a[i] < a[y])) **then**

**begin**

b[k] := a[i];

i := i + 1;

**end**

**else**

**begin**

b[k] := a[y];

y := y + 1;

**end**;

**for** k := p **to** q **do**

a[k] := b[k];

**end**;

**procedure** sortprost(**var** a: ms; p, q: integer);

**begin**

**if** p < q **then**

**begin**

sortprost(a, p, (p + q) **div** 2);

sortprost(a, (p + q) **div** 2 + 1, q);

sliynie(a, p, q);

**end**;

**end**;

**procedure** sortLoc;

**var**

a: ms;

n, i, x: integer;

mal, inp: text;

**begin**

clrscr;

writeln('Наш массив: ');

Assign(mal, 'D:\мал.txt');

Reset(mal);

n := 0;

**repeat**

readln(mal, x);

inc(n);

**until** eof(mal);

Reset(mal);

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

**begin**

readln(mal, a[i]);

writeln(a[i]);

**end**;

close(mal);

writeln;

sortprost(a, 1, n);

writeln('Отсортированный массив: ');

Assign(out, 'D:\output.txt');

ReWrite(out);

**if** xz = 1 **then**

**begin**

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

**begin**

writeln(out, a[i]);

writeln(a[i]);

**end**;

**end**;

**if** xz = 2 **then**

**begin**

**for** i := n **downto** 1 **do**

**begin**

writeln(out, a[i]);

writeln(a[i]);

**end**;

**end**;

close(out);

**end**;

**begin**

**repeat**

clrScr;

writeln('1. Сортировка подсчетом');

writeln('2. Сортировка слиянием');

writeln('3. Выход');

write('Выберите пункт: ');

readln(xz);

**case** xz **of**

1:

**begin**

clrscr;

writeln('1. Отсортировать по возрастанию');

writeln('2. Отсортировать по убыванию');

write('Выберите пункт: ');

readln(xz);

**case** xz **of**

1:

**begin**

clrScr;

sortObw;

readln;

**end**;

2:

**begin**;

clrScr;

sortObw;

readln;

**end**;

**end**;

**end**;

2:

**begin**

clrscr;

writeln('1. Отсортировать по возрастанию');

writeln('2. Отсортировать по убыванию');

write('Выберите пункт: ');

readln(xz);

**case** xz **of**

1:

**begin**

clrScr;

sortLoc;

readln;

**end**;

2:

**begin**;

clrScr;

sortLoc;

readln;

**end**;

**end**;

**end**;

3: **exit**;

**end**;

**until** xz = 3;

**end**.

**6. Результат выполнения программы**

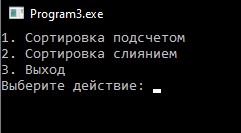


Рисунок 6 – главное меню программы.

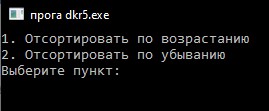


Рисунок 7 – выполнение программы.

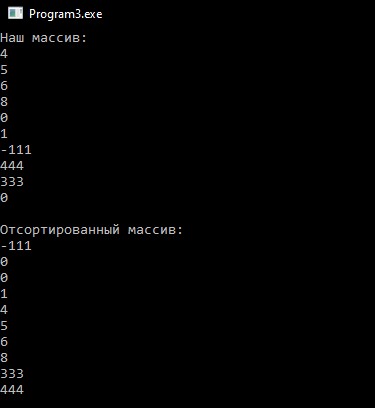


Рисунок 8 – результат выполнения программы.

**7. Вывод**

Цель данной контрольной работы, «Получить базовые сведения о наиболее известных алгоритмах сортировки, изучить принципы работы с текстовыми файлами.», была успешно достигнута.

Были изучены различные методы, а именно – алгоритм подсчетом и алгоритм слияния. После изучения и понимания мы радовались и хотели разделить наше счастье со всеми. Ведь не просто так говорится: “Когда ты счастлив – ты хочешь, чтобы все вокруг были счастливы”. С поставленными задачами мы справились успешно, поэтому домашнюю контрольную номер 5 можно считать успешно выполненной.