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-205-52-00

Останин Кирилл Игоревич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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

**Формулировка задания**:

**Вариант 13.**

1. Реализовать сортировку данных с помощью алгоритма выбором.
2. Реализовать сортировку данных с помощью пирамидального 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 вводных данных.

**Описание алгоритма**:

1. Считать входные данные из текстового файла. Каждая строка файла представляет собой один элемент данных, который нужно отсортировать.

2. Реализовать функцию сравнения (компаратор), которая будет определять порядок сортировки элементов. Эта функция должна передаваться в подпрограмму сортировки.

3. Реализовать алгоритм сортировки выбором:

- Пройти по всем элементам массива.

- Найти минимальный элемент в оставшейся части массива.

- Поменять местами минимальный элемент с текущим элементом.

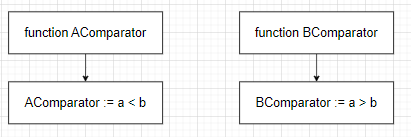
4. Реализовать алгоритм пирамидальной сортировки:

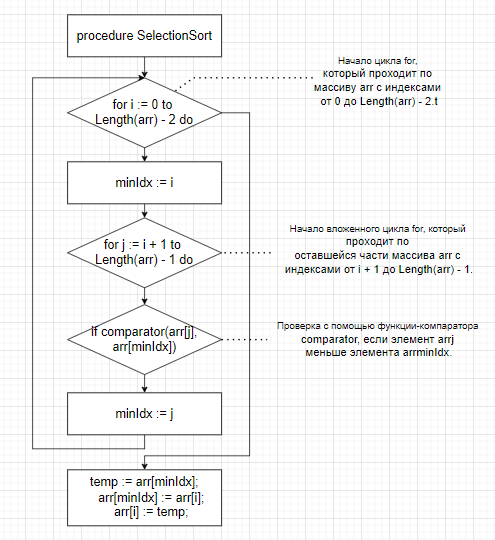
- Построить из массива пирамиду (кучу).

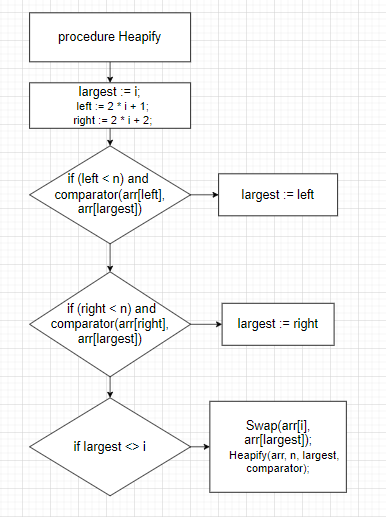
- Постепенно извлекать элементы из пирамиды, что приводит к упорядочиванию элементов.

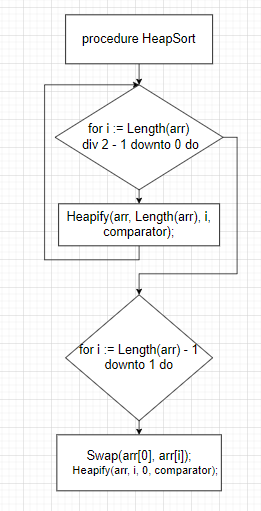
5. Подготовить варианты вводных данных для демонстрации работы программы, где каждая строка текстового файла будет содержать элементы данных для сортировки.

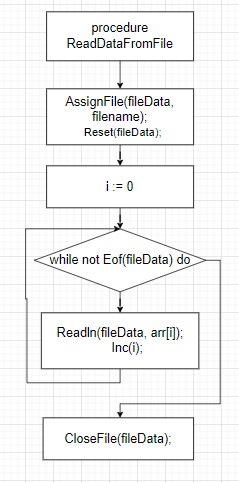
**Схема алгоритма с комментариями:**

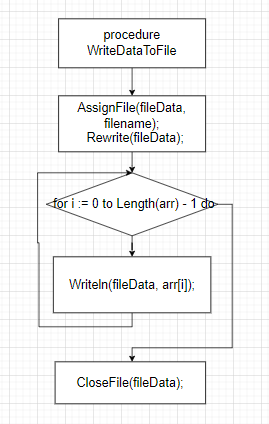


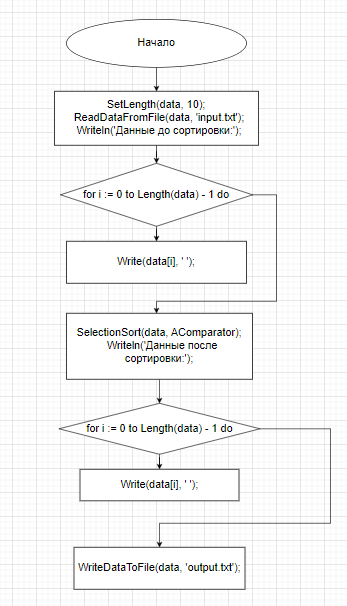












**Код программы:**

**type**

Comparator = **function**(a, b: integer): boolean;

**function** AComparator(a, b: integer): boolean;

**begin**

AComparator := a < b;

**end**;

**function** BComparator(a, b: integer): boolean;

**begin**

BComparator := a > b;

**end**;

**procedure** SelectionSort(**var** arr: **array of** integer; comparator: Comparator);

**var**

i, j, minIdx, temp: integer;

**begin**

**for** i := 0 **to** Length(arr) - 2 **do**

**begin**

minIdx := i;

**for** j := i + 1 **to** Length(arr) - 1 **do**

**begin**

**if** comparator(arr[j], arr[minIdx]) **then**

minIdx := j;

**end**;

temp := arr[minIdx];

arr[minIdx] := arr[i];

arr[i] := temp;

**end**;

**end**;

**procedure** Heapify(**var** arr: **array of** integer; n, i: integer; comparator: Comparator);

**var**

largest, left, right: integer;

**begin**

largest := i;

left := 2 \* i + 1;

right := 2 \* i + 2;

**if** (left < n) **and** comparator(arr[left], arr[largest]) **then**

largest := left;

**if** (right < n) **and** comparator(arr[right], arr[largest]) **then**

largest := right;

**if** largest <> i **then**

**begin**

Swap(arr[i], arr[largest]);

Heapify(arr, n, largest, comparator);

**end**;

**end**;

**procedure** HeapSort(**var** arr: **array of** integer; comparator: Comparator);

**var**

i: integer;

**begin**

**for** i := Length(arr) **div** 2 - 1 **downto** 0 **do**

Heapify(arr, Length(arr), i, comparator);

**for** i := Length(arr) - 1 **downto** 1 **do**

**begin**

Swap(arr[0], arr[i]);

Heapify(arr, i, 0, comparator);

**end**;

**end**;

**procedure** ReadDataFromFile(**var** arr: **array of** integer; filename: string);

**var**

fileData: TextFile;

i: integer;

**begin**

AssignFile(fileData, filename);

Reset(fileData);

i := 0;

**while not** Eof(fileData) **do**

**begin**

Readln(fileData, arr[i]);

Inc(i);

**end**;

CloseFile(fileData);

**end**;

**procedure** WriteDataToFile(arr: **array of** integer; filename: string);

**var**

fileData: TextFile;

i: integer;

**begin**

AssignFile(fileData, filename);

Rewrite(fileData);

**for** i := 0 **to** Length(arr) - 1 **do**

Writeln(fileData, arr[i]);

CloseFile(fileData);

**end**;

**var**

data: **array of** integer;

i: integer;

**begin**

SetLength(data, 10);

ReadDataFromFile(data, 'input.txt');

Writeln('Данные до сортировки:');

**for** i := 0 **to** Length(data) - 1 **do**

Write(data[i], ' ');

Writeln;

SelectionSort(data, AComparator);

Writeln('Данные после сортировки:');

**for** i := 0 **to** Length(data) - 1 **do**

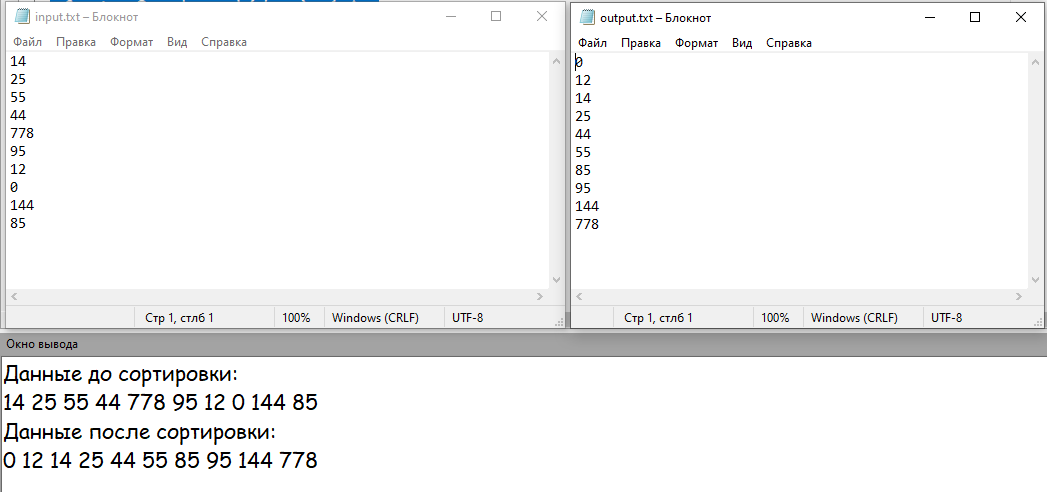
Write(data[i], ' ');

Writeln;

WriteDataToFile(data, 'output.txt');

**end**.

**Результат выполнения программы:**



**Вывод:**

В ходе выполнения контрольной домашней работы были улучшены навыки в организации времени и постановке приоритетов выполнения задач.

Несмотря на некоторые сложности с оформлением и ограничением по времени, задание было выполнено. Получен положительный опыт и знания. Эта работа помогла развить навыки работы в условиях ограниченного времени и глубже понять изучаемый материал.

Кроме этого, стало ясно, что важно создавать чёткий план работы перед началом выполнения задания. Это позволяет более организованно подходить к решению задач и не тратить время на ненужные действия.

В целом, контрольная работа была полезным занятием, которое помогло мне развить навыки решения задач в ограниченное время и углубить знания в предмете.