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-206-52-00

Федяев Алексей Сергеевич

Преподаватель:

Кузьминых Ангелина Владимировна

Киров

2023

**Цель работы (Вариант 11)**: 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.

Задание 2: реализовать сортировку данных с помощью алгоритма слияния.

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

Задание 4: считывание и вывод данных необходимо производить из текстового файла.

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

**Описание алгоритма 1:**

* Ввод данных из файла
* Выполнение сортировки подсчётом
* Вывод данных в файл

Описание алгоритма 2:

* Ввод данных из файла
* Выполнение сортировки слиянием
* Вывод данных в файл

Описание алгоритма 3:

* Ввод данных из файлов
* Сравнение массивов
* Вывод результатов

**Блок схема задания 1:**



**Блок схема задания 2:**



**Алгоритм решения задачи 3:**



**Код программы 1:**

**program** lab5;

**const** N = 10000;

**type** massiv=**array**[1..N] **of** integer;

**var** i: integer;

input,out1,out2: text;

a,b,long\_arr: massiv;

**function** TxtToArr(inputfile:string): massiv;

**var** i: integer;

input:text;

**begin**

assign(input,inputfile);

reset(input);

i:=1;

**while not** EOF(input) **do**

**begin**

readln(input, result[i]);

i += 1;

**end**;

close(input);

**end**;

**procedure** ArrToTxt(a:massiv; outputfile:string);

**var** i,len:integer;

output:text;

**begin**

assign(output,outputfile);

rewrite(output);

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

writeln(output,a[i]);

close(output);

**end**;

**procedure** comparator(a, b:massiv);

**var** i,len1,len2:integer;

t:boolean;

**begin**

t:=True;

**for** i:=1 **to** len1 **do**

**if** a[i] <> b[i] **then**

**begin**

t:=False;

**break**;

**end**;

**if** t **then**

writeln('Массивы одинаковы')

**else**

writeln('Массивы различаются')

**end**;

**function** counting\_sort(a:massiv): massiv;

**var** i,j,k:integer;

f:boolean;

**begin**

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

result[i]:=-1;

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

**begin**

k:=0;

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

**if** a[j]<a[i] **then** k:=k+1;

result[k+1]:=a[i];

**end**;

**for** i:=2 **to** n **do**

**if** result[i]=-1 **then**

result[i]:=result[i-1];

**end**;

**procedure** Merge(**var** A: massiv; first, last: integer);

**var** middle, start, final , j: integer;

mas: massiv;

**begin**

middle:=(first+last) **div** 2;

start:=first;

final:=middle+1;

**for** j:=first **to** last **do**

**if** (start <= middle) **and** ((final>last) **or** (A[start]<A[final])) **then**

**begin**

mas[j]:=A[start];

start += 1;

**end**

**else**

**begin**

mas[j]:=A[final];

final += 1;

**end**;

**for** j:=first **to** last **do** A[j]:=mas[j];

**end**;

**procedure** MergeSort(**var** A: massiv; first, last: integer);

**begin**

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

**begin**

MergeSort(A, first, (first+last) **div** 2);

MergeSort(A, (first+last) **div** 2+1, last);

Merge(A, first, last);

**end**;

**end**;

**Begin**

Long\_arr := TxtToArr('long\_arr.txt');

a := counting\_sort(long\_arr);

b:=long\_arr;

MergeSort(b,1,n);

Writeln('Массивы отсортированы');

comparator(a,b);

ArrToTxt(long\_arr,'dasfds');

ArrToTxt(a,'count\_sorted.txt');

ArrToTxt(b,'merge\_sorted.txt');

**end**.

**Вывод:**

В ходе выполнения лабораторной работы получены базовые сведения о наиболее известных алгоритмах сортировки, изучены принципы работы с текстовыми файлами, получены навыки реализации алгоритмов сортировки на языке программирования Pascal. Однако в процессе работы выявились следующие проблемы:

1. Непроверенный ввод данных. Если в файле будут присутствовать нечисловые значения это приведет к ошибке выполнения программы.
2. Переполнение стека вызовов. Сортировка слиянием – рекурсивный алгоритм, при неправильной реализации которого происходит переполнение стека вызовов.