Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(ПНИПУ)

ОТЧЁТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ

по теме:

ВНЕШНИЕ СОРТИРОВКИ

Выполнила: студентка группы РИС-22-1б

Черкасова А.А\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверил: доцент кафедры ИТАС

Полякова О.А\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_/\_\_\_\_\_\_\_/\_\_\_\_\_г.

Пермь 2023

**Содержание**

[**Введение** 3](#_Toc133680724)

[**Анализ задачи** 3](#_Toc133680725)

[**Приложение А** 4](#_Toc133680726)

[**Приложение Б** 6](#_Toc133680727)

# **Введение**

**Постановка задачи:** Произвести сортировку массива с применение

**Цель:** Отсортировать массив.

Для достижения поставленной цели, необходимо решить следующие **задачи**:

* Провести анализ задачи
* Реализовать задачу на языке С++
* Составить блок-схему

# **Анализ задачи**

Сначала задача разбивается на несколько подзадач меньшего размера. Затем эти подзадачи решаются с помощью рекурсивного вызова функции. Последним действием является комбинирование решений каждой подзадачи (слияние), которое приводит к искомому результату.

1. Деление на серии

2. Попарное слияние серий с одновременной сортировкой

3. Повторение шагов (1-2) до получения единственной серии, то есть полностью отсортированных данных

4. Условием выхода из рекурсии является получение полностью отсортированного множества (файла)

## **Приложение А**

**Листинг программы**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

void Distribution(string path1, string path2, string main\_path) {

ofstream f1, f2;

ifstream mainf;

string temp, cur;

int itemp, icur;

f1.open(path1);

f2.open(path2);

mainf.open(main\_path);

if (!f1.is\_open() || !f2.is\_open()) { cout << "Error" << endl; }

else {

bool firstfile = true;

mainf >> cur;

while (!mainf.eof()) {

if (firstfile) {

f1 << cur << "\n";

temp = cur;

mainf >> cur;

if (mainf.eof()) {

f1 << "/" << "\n";

return;

}

itemp = stoi(temp);

icur = stoi(cur);

while (itemp <= icur) {

if (mainf.eof()) {

f1 << "/" << "\n";

return;

}

f1 << cur << "\n";

temp = cur;

mainf >> cur;

itemp = stoi(temp);

icur = stoi(cur);

}

f1 << "/" << "\n";

firstfile = false;

}

else {

f2 << cur << "\n";

temp = cur;

mainf >> cur;

if (mainf.eof()) {

f2 << "/" << "\n";

return;

}

itemp = stoi(temp);

icur = stoi(cur);

while (itemp <= icur) {

if (mainf.eof()) {

f2 << "/" << "\n";

return;

}

f2 << cur << "\n";

temp = cur;

mainf >> cur;

itemp = stoi(temp);

icur= stoi(cur);

}

f2 << "/" << "\n";

firstfile = true;

}

}

}

f1.close();

f2.close();

mainf.close();

}

void Merge(string path1, string path2, string main\_path) {

Distribution(path1, path2, main\_path);

ifstream f1, f2;

ofstream mainf;

string templ, temp2;

int itempl, itemp2;

f1.open(path1);

f2.open(path2);

mainf.open(main\_path);

if (!f1.is\_open() || !f2.is\_open()) { cout << "Error" << endl; }

else {

int i, j = 0;

while (!(f1.eof())) {

f1 >> templ;

f2 >> temp2;

if (temp2 == "") return;

while (templ != "/" || temp2 != "/") {

if (templ == "/") {

mainf << temp2 << "\n";

f2 >> temp2;

}

else if (temp2 == "/") {

mainf<< templ << "\n";

f1 >> templ;

}

else {

itempl = stoi(templ);

itemp2 = stoi(temp2);

if (itempl <= itemp2) {

mainf << templ << "\n";

f1 >> templ;

}

else {

mainf << temp2 << "\n";

f2 >> temp2;

}

}

}

}

}

f1.close();

f2.close();

mainf.close();

Merge(path1, path2, main\_path);

}

int main() {

srand(time(NULL));

int size, current;

string result = "RESULT.txt", buff = "buffer.txt", datafile = "data.txt", temp;

ofstream data;

ifstream res;

cout << "Size of an array: ";

cin >> size;

cout << "unsorted data" << endl;

//заполнение файла с данными случайными значениями

data.open(datafile);

for (int i = 0; i < size; i++) {

current = rand() % 100;

cout << current << " ";

data << current << "\n";

}

data.close();

cout << endl << endl;

Merge(result, buff, datafile); //вызов ункчии сортировки

cout << "sorted data" << endl;

//вывод в консоль данных в результирующем файле

res.open(result);

for (int i = 0; i < size; i++) {

res >> temp;

cout << temp << " ";

}

res.close();

cout << endl;

return 0;

}

# **Приложение Б**

**Результаты выполнения программы**



