Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий

Кафедра вычислительной техники

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №7**

По дисциплине “Алгоритмы и структуры данных”

Вариант 21

|  |  |  |
| --- | --- | --- |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_ | Матковский И.В. |
|  | подпись, дата |  |
| Студент КИ20-07Б, 032052463 | \_\_\_\_\_\_\_\_\_\_ | Базаров А.С. |
|  | подпись, дата |  |
|  |  |  |
|  |  |  |

Красноярск 2021

**СОДЕРЖАНИЕ**

[1 Задание 4](#_Toc81912682)

[2 Исходный код 4](#_Toc81912683)

[3 Результаты 5](#_Toc81912684)

1. Задание

Вариант: a.

Используя один из описанных в лекции алгоритмов внешней сортировки, отсортировать [массив](https://e.sfu-kras.ru/mod/glossary/showentry.php?eid=109349&displayformat=dictionary). Исходный массив целых чисел считывается из входного файла. Количество элементов в массиве может превышать 1012.

1. Исходный код

Программа с внешней сортировкой слиянием (к программе прикреплен файл с массивом чисел):

#pragma warning(disable : 4996)

#include <clocale>

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

//Описание функции сортировки простым слиянием

void Simple\_Merging\_Sort(char\* name) {

int a1, a2, k, i, j, kol, tmp;

FILE\* f, \* f1, \* f2;

kol = 0;

if ((f = fopen(name, "r")) == NULL)

printf("\nИсходный файл не может быть прочитан...");

else {

while (!feof(f)) {

fscanf(f, "%d", &a1);

kol++;

}

fclose(f);

}

k = 1;

while (k < kol) {

f = fopen(name, "r");

f1 = fopen("smsort\_1", "w");

f2 = fopen("smsort\_2", "w");

if (!feof(f)) fscanf(f, "%d", &a1);

while (!feof(f)) {

for (i = 0; i < k && !feof(f); i++) {

fprintf(f1, "%d ", a1);

fscanf(f, "%d", &a1);

}

for (j = 0; j < k && !feof(f); j++) {

fprintf(f2, "%d ", a1);

fscanf(f, "%d", &a1);

}

}

fclose(f2);

fclose(f1);

fclose(f);

f = fopen(name, "w");

f1 = fopen("smsort\_1", "r");

f2 = fopen("smsort\_2", "r");

if (!feof(f1)) fscanf(f1, "%d", &a1);

if (!feof(f2)) fscanf(f2, "%d", &a2);

while (!feof(f1) && !feof(f2)) {

i = 0;

j = 0;

while (i < k && j < k && !feof(f1) && !feof(f2)) {

if (a1 < a2) {

fprintf(f, "%d ", a1);

fscanf(f1, "%d", &a1);

i++;

}

else {

fprintf(f, "%d ", a2);

fscanf(f2, "%d", &a2);

j++;

}

}

while (i < k && !feof(f1)) {

fprintf(f, "%d ", a1);

fscanf(f1, "%d", &a1);

i++;

}

while (j < k && !feof(f2)) {

fprintf(f, "%d ", a2);

fscanf(f2, "%d", &a2);

j++;

}

}

while (!feof(f1)) {

fprintf(f, "%d ", a1);

fscanf(f1, "%d", &a1);

}

while (!feof(f2)) {

fprintf(f, "%d ", a2);

fscanf(f2, "%d", &a2);

}

fclose(f2);

fclose(f1);

fclose(f);

k \*= 2;

}

remove("smsort\_1");

remove("smsort\_2");

}

int main()

{

setlocale(LC\_ALL, "rus");

fstream A;

fstream B;

int a;

char Aa[8] = "A.txt";

char Bb[8] = "B.txt";

B.open(Bb, ios :: out);

B << "";

B.close();

A.open(Aa);

B.open(Bb);

cout << "Неотсортированный массив" << endl;

while (A >> a) {

cout << a << " ";

B << a << " ";

}

A.seekg(0);

A.close();

B.close();

Simple\_Merging\_Sort(Bb);

cout << endl;

B.open(Bb);

cout << "Отсортированный массив" << endl;

int count = 0;

while (B >> a) {

count++;

cout << a << " ";

}

B.seekg(0);

B.close();

cout << endl;

int ind, i = 0, el;

Return:

cout << "Введите индекс элемента:";

cin >> ind;

if (ind > count - 1) { cout << "Индекс больше массива!" << endl; goto Return; }

B.open(Bb);

while (i < ind) { B >> el; i++; }

B >> el;

cout << "Элемент по индексу " << ind << " в отсортированном массиве = " << el;

B.seekg(0);

B.close();

return 0;

}

1. Теоретические оценки сложности

Теоретическая сложность данной программы при внешней сортировке слиянием составляет , (где n – массив чисел) так как мы выводим массив, сортируем его и выводим уже отсортированный.

1. Результаты

Результаты выполнения алгоритма приведены в таблице 1

Таблица 1

|  |  |
| --- | --- |
| Количество элементов | Внешняя сортировка слиянием, время в мс. |
| 100 | 0,66 |
| 5000 | 52,59 |
| 100000 | 1351,29 |