



МИНОРБНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение высшего
образования*

«МИРЭА - Российский технологический университет»

РТУ МИРЭА

Отчёт по выполнению практического задания № 4

Тема:

«Алгоритмы внешних сортировок»

Дисциплина: «Структуры и алгоритмы обработки данных»

Выполнил студент:

Фамилия И.О.

Фамилия И.О.

Группа:

AAAA-00-00

Номер группы

Москва 2023

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	3
1.1 Цель работы.....	3
1.2 Задание 1.....	3
1.3 Задание 2.....	3
1.4 Индивидуальный вариант.....	4
2 ЗАДАНИЕ 1.....	5
2.1 Реализация функции сортировки прямого слияния.....	5
2.3 Тестирование алгоритма сортировки данных $n = 8, 16, 32$	8
2.4 Выводы об эффективности алгоритма.....	8
2 ЗАДАНИЕ 2.....	9
3.1 Реализация функции сортировки естественного слияния.....	9
2.3 Тестирование алгоритма сортировки данных $n = 8, 16, 32$	12
2.4 Выводы об эффективности алгоритма.....	12
4 ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ.....	13

1 ПОСТАНОВКА ЗАДАЧИ

1.1 Цель работы

Освоить приёмы сортировки данных из файлов.

1.2 Задание 1

Разработать программу и применить алгоритм внешней сортировки прямого слияния к сортировке файла данных индивидуального варианта.

1) Реализовать функцию сортировки (возможно, с вспомогательными функциями) и основную подпрограмму `main`.

2) Отладить программу, протестировать на примере.

3) Предварительно подготовить файл данных в соответствии с вариантом (не менее 32 записей).

4) Адаптировать программу для сортировки файла с записями, протестировать на подготовленном ранее файле.

5) Определить практическую сложность алгоритма для файлов с увеличивающимся количеством записей (8, 16, 32). Сформировать таблицу результатов, указав количество записей и время сортировки.

1.3 Задание 2

Разработать программу и применить алгоритм сортировки естественного слияния к сортировке файла с данными варианта (файл уже должен быть подготовлен в задании 1).

1) Реализовать функцию сортировки (возможно, с вспомогательными функциями) и основную подпрограмму `main`.

2) Отладить программу, протестировать на примере из п.4.

3) Адаптировать программу для сортировки файла с записями, протестировать на подготовленном ранее файле.

4) Сформировать таблицу результатов, указав количество записей и время сортировки.

1.4 Индивидуальный вариант

Таблица 1 - Индивидуальный вариант структуры записи файла

17	Справочник туриста. Турагентство предлагает услуги: Страна, Город, Условия проживания (Отель-звезды, Автобус и т. д.), Условия проезда, Экскурсионное обслуживание, Сервис принимающей стороны, Стоимость путевки.
----	--

2 ЗАДАНИЕ 1

2.1 Реализация функции сортировки прямого слияния

Для тестирования функции сортировки прямого слияния разработаем алгоритм прямого слияния. Код представлен на рис 1, 2. Алгоритм разделяет данные по порциям, которые сливает в один файл, постепенно увеличивая размер порции. Поскольку в задании необходимо сравнивать записи по первому ключу, то достаточно просто сравнивать строки записей напрямую.

```
void mergeFiles(int chunkSize, int n)
{
    ifstream array("A.txt");
    ofstream lArr("B.txt");
    ofstream rArr("C.txt");

    int cl = 0;
    int cr = 0;
    bool q = true;
    string line;
    getline(array, line);
    int j = 0;
    while (j < n)
    {
        if (cl < chunkSize && q)
        {
            cl++;
            j++;
            lArr << line << endl;
            getline(array, line);
        }
        else
        {
            cl = 0;
            q = false;
        }
        if (cr < chunkSize && !q)
        {
            cr++;
            j++;
            rArr << line << endl;
            getline(array, line);
        }
        else
        {
            cr = 0;
            q = true;
        }
    }
}
```

Рисунок 1 - Начало алгоритма сортировки

```

.....getline(lSArr, l);
.....getline(rSArr, r);
.....while (ch < n / 2)
.....{
.....    int lq = 0;
.....    int rq = 0;

.....    while (lq < chunkSize && rq < chunkSize)
.....    {
.....        if (l <= r)
.....        {
.....            Arr << l << endl;
.....            lq++;
.....            getline(lSArr, l);
.....        }
.....        else
.....        {
.....            Arr << r << endl;
.....            rq++;
.....            getline(rSArr, r);
.....        }
.....    }

.....    while (lq < chunkSize)
.....    {
.....        Arr << l << endl;
.....        lq++;
.....        getline(lSArr, l);
.....    }
.....    while (rq < chunkSize)
.....    {
.....        Arr << r << endl;
.....        rq++;
.....        getline(rSArr, r);
.....    }

.....    ch += chunkSize;
.....}

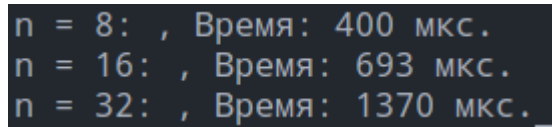
```

Рисунок 2 - Продолжение алгоритма сортировки

Блок-схема алгоритма сортировки представлена на рис. 3.

2.3 Тестирование алгоритма сортировки данных $n = 8, 16, 32$

Тестирование алгоритма представлено на рисунке 4.



The image shows a screenshot of a terminal window with a dark background and light-colored text. It displays three lines of test results for a sorting algorithm with different input sizes (n). The first line shows n = 8 with a time of 400 microseconds. The second line shows n = 16 with a time of 693 microseconds. The third line shows n = 32 with a time of 1370 microseconds.

n	Время (мкс.)
8	400
16	693
32	1370

Рисунок 4 - Результаты
тестирования

2.4 Выводы об эффективности алгоритма

Анализируя результаты сортировки, можно сделать вывод, что время выполнения алгоритма растёт пропорционально количеству записей. При этом сам алгоритм является быстрым и эффективным.

2 ЗАДАНИЕ 2

3.1 Реализация функции сортировки естественного слияния

Для тестирования функции сортировки прямого слияния разработаем алгоритм прямого слияния. Код представлен на рис 5, 6. Алгоритм разделяет данные на упорядоченные серии, которые сливает в один файл. Продолжается рекурсивно, до тех пор, пока количество серии не будет равно 1. Поскольку в задании необходимо сравнивать записи по первому ключу, то достаточно просто сравнивать строки записей напрямую.

```
void natural_merge_sort(const string &input_filename, const string &output_filename)
{
    ifstream input(input_filename);
    ofstream output(output_filename);
    vector<string> block;
    while (true)
    {
        block.clear();
        for (int i = 0; i < BLOCK_SIZE; i++)
        {
            string line;
            getline(input, line);
            if (line.empty())
            {
                break;
            }
            block.push_back(line);
        }
        sort_block(block);
        if (block.size() < BLOCK_SIZE)
        {
            for (const string &line : block)
            {
                output << line << endl;
            }
            break;
        }
        string temp_filename = "temp.txt";
        ofstream temp(temp_filename);
        for (const string &line : block)
        {
            temp << line << endl;
        }
        temp.close();
        natural_merge_sort(temp_filename, output_filename);
        remove(temp_filename.c_str());
    }
    input.close();
    output.close();
}
```

Рисунок 5 - Основной алгоритм сортировки

```

void merge_blocks(const vector<string> &block1,
const vector<string> &block2, vector<string> &output)
{
    int i = 0, j = 0;
    while (i < block1.size() && j < block2.size())
    {
        if (block1[i] < block2[j])
        {
            output.push_back(block1[i++]);
        }
        else
        {
            output.push_back(block2[j++]);
        }
    }
    while (i < block1.size())
    {
        output.push_back(block1[i++]);
    }
    while (j < block2.size())
    {
        output.push_back(block2[j++]);
    }
}

```

Рисунок 6 - Алгоритм слияния серий

Блок-схема алгоритма сортировки представлена на рис. 7.

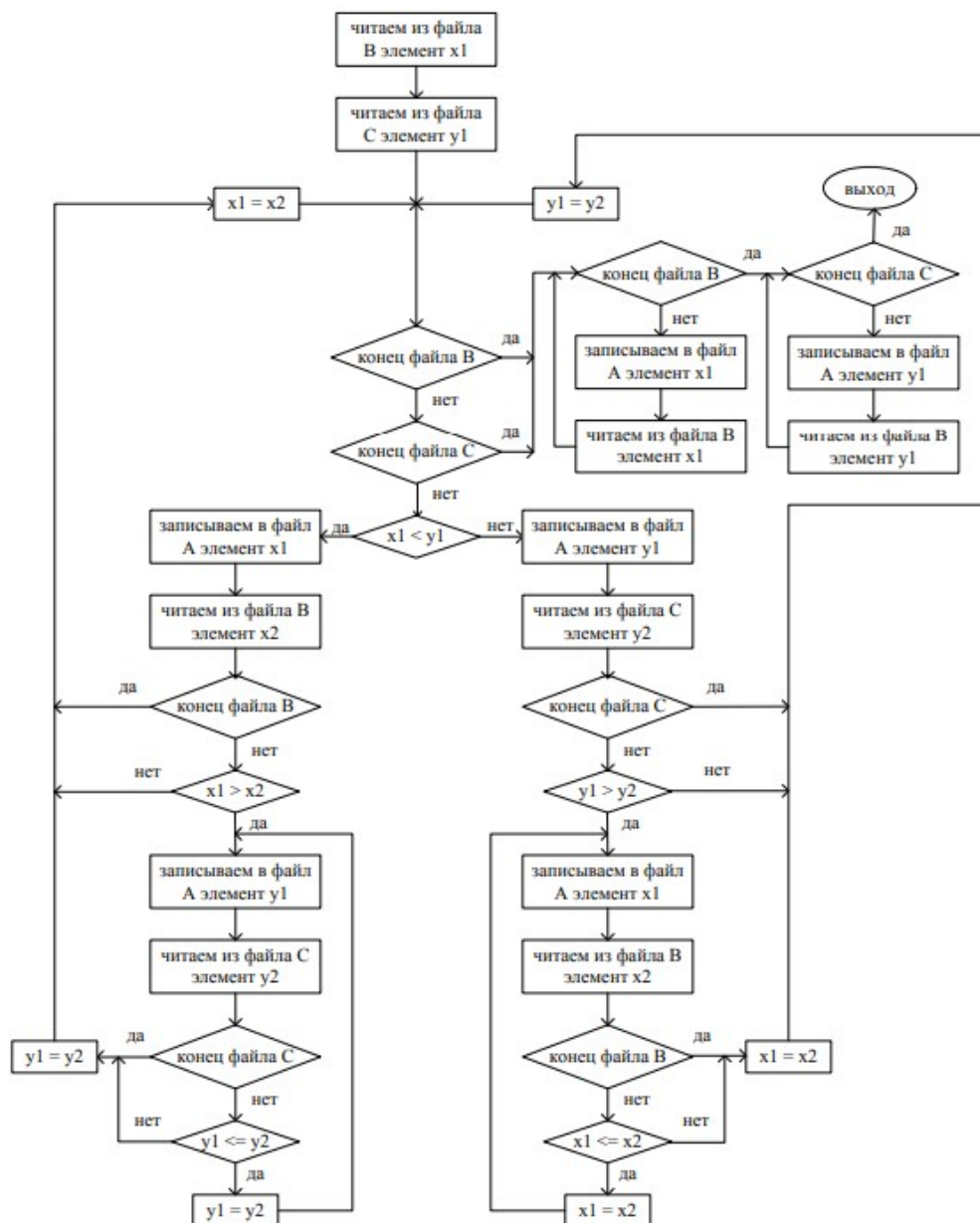


Рисунок 7 - Блок-схема алгоритма сортировки естественного слияния

2.3 Тестирование алгоритма сортировки данных $n = 8, 16, 32$

Тестирование алгоритма представлено на рисунке 8. Результаты в файле до и после на рис. 9, 10 ($n = 8$)

```
n = 8: , Время: 61 мкс.  
n = 16: , Время: 94 мкс.  
n = 32: , Время: 136 мкс.
```

Рисунок 8 - Результаты тестирования

```
1 Турция Анталия 4 Авиаперелет Включено(обзорнаяэкскурсияпогороду) Трансфер,услугигида 30000  
2 Египет Шарм-эль-Шейх 5звезд Авиаперелет Невключено Трансфер,визоваяподдержка 40000  
3 ОАЭ Дубай 3 Автобусныйтур Невключено Трансфер,консультациипоэкскурсиям 25000  
4 Испания Барселона Хостел Железнодорожныйтранспорт Включено(посещениеПаркаГуэль) Трансфер,круглосуточнаяподдержка 35000  
5 Таиланд Пхукет Бунгало Авиаперелет Включено(поездканаострова) Трансфер,услугигида 50000  
6 Италия Рим Апартаменты Авиаперелет Невключено Трансфер,бронированиебилетовнадостопримечательности 45000  
7 Греция Афины 4 Паромноесообщение Включено(экскурсиянаАкрополь) Трансфер,услугигида 38000  
8 Франция Париж 3 Автобусныйтур Невключено Трансфер,визоваяподдержка 30000
```

Рисунок 9 - Содержимое файла до сортировки

```
1 Греция Афины 4 Паромноесообщение Включено(экскурсиянаАкрополь) Трансфер,услугигида 38000  
2 Египет Шарм-эль-Шейх 5звезд Авиаперелет Невключено Трансфер,визоваяподдержка 40000  
3 Испания Барселона Хостел Железнодорожныйтранспорт Включено(посещениеПаркаГуэль) Трансфер,круглосуточнаяподдержка 35000  
4 Италия Рим Апартаменты Авиаперелет Невключено Трансфер,бронированиебилетовнадостопримечательности 45000  
5 ОАЭ Дубай 3 Автобусныйтур Невключено Трансфер,консультациипоэкскурсиям 25000  
6 Таиланд Пхукет Бунгало Авиаперелет Включено(поездканаострова) Трансфер,услугигида 50000  
7 Турция Анталия 4 Авиаперелет Включено(обзорнаяэкскурсияпогороду) Трансфер,услугигида 30000  
8 Франция Париж 3 Автобусныйтур Невключено Трансфер,визоваяподдержка 30000
```

Рисунок 10 - Содержимое файла после сортировки

2.4 Выводы об эффективности алгоритма

Анализируя результаты сортировки, можно сделать вывод, что время выполнения алгоритма растёт пропорционально количеству записей. При этом сам алгоритм более оптимизирован для больших значений n и значительно быстрее чем алгоритм прямого слияния.

4 ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ

1. Структуры и алгоритмы обработки данных : Лекционные материалы / Рысин М. Л. МИРЭА — Российский технологический университет, 2022/23. – 77 с.

2. Merge Sort – Data Structure and Algorithms Tutorialss [Электронный ресурс]. URL: <https://www.geeksforgeeks.org/merge-sort/> (дата обращения 11.03.2024).