

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«МИРЭА - Российский технологический университет»**

**РТУ МИРЭА**

Институт информационных технологий

Кафедра математического обеспечения и стандартизации

информационных технологии

**ОТЧЕТ**

**ПОПРАКТИЧЕСКОЙРАБОТЕ№ 4**

«Алгоритмы внешних сортировок»

**по дисциплине**

**«**СТРУКТУРЫ И АЛГОРИТМЫ ОБРАБОТКИ ДАННЫХ**»**

Выполнил студент группы ИКБО-02-21 Хитров Н. С.

Принял преподаватель   Рысин М.Л.

|  |  |  |
| --- | --- | --- |
| Практическая работа выполнена | « » \_\_\_\_\_\_ 2022 г. |  |
|  |  |  |
| «Зачтено» | « » \_\_\_\_\_\_ 2022 г. | |

Москва 2022

# ПОСТАНОВКА ЗАДАЧИ

ЦЕЛЬ РАБОТЫ:

Освоить приёмы сортировки данных из файлов.

ЗАДАЧИ:

1. Для каждого алгоритма представить его словесное описание и блок-схему, код функций сортировки, результаты тестирования.
2. Отразить результаты сортировки файла с записями в соответствии с заданиями 1 и 2.
3. Сделать выводы об эффективности алгоритмов на основе полученных практических замеров времени выполнения.

Индивидуальный вариант №15:

Справочник владельца видеотеки. О каждом видеофильме хранятся данные: Название, Студия, Жанр, Год выпуска, Режиссер, Исполнители главных ролей (не более 10): фамилия

# ЗАДАНИЕ №1

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

## Постановка задачи

1. Реализовать функцию сортировки (возможно, с вспомогательными функциями) и основную подпрограмму main.
2. Отладить программу, протестировать на предоставленном примере.
3. Предварительно подготовить файл данных в соответствии с вариантом.
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

## Алгоритм сортировки прямого слияния

Сначала устанавливается размер порции сливаемых данных. Затем для каждой порции считываются по одной записи из файлов В и С. После происходит слияние: меньшая запись записывается в файл А, и считывается очередная запись из того файла, запись которого была переписана в файл А. После размер порции увеличивается. Сортировка заканчивается, когда размер порции равен изначальному количеству строк.

Пример – демонстрация работы алгоритма прямого слияния на массиве ключей.

Пусть файл А содержит данные, которые подлежат сортировке:

8 2 13 4 15 6 9 11 3 7 5 10 1 12 14

Для реализации алгоритма будем использовать два файла, в которые будем разливать данные файла: файл В и файл С.

Сначала разбиваем по одному элементу:

B: 8 13 15 9 3 5 1 14

C: 2 4 6 11 7 10 12

Сливаем в упорядоченные двойки:

А: 2 8 4 13 6 15 9 11 3 7 5 10 1 12 14 2

Разливаем по два:

B: 2 8 6 15 3 7 1 12

C: 4 13 9 11 5 10 14

Сливаем в упорядоченные четверки и т. д. пока длина порции не станет равной длине массива.

Блок-схема алгоритма представлена на рисунке 1.

Рисунок 1 – Блок-схема сортировки прямого слияния

Алгоритм использует следующую структуру:

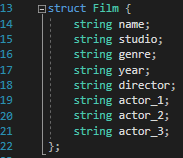


Рисунок 2 – Используемая структура

Рисунок 3 иллюстрирует часть кода, где реализована фаза разделения алгоритма сортировки.

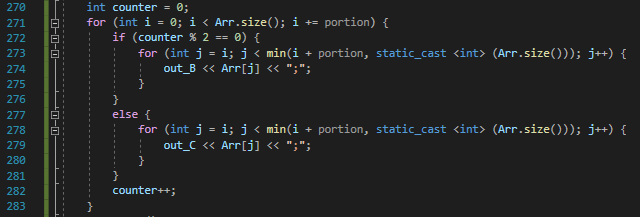


Рисунок 3 – Фаза разделения

Фаза слияния выглядит следующим образом:

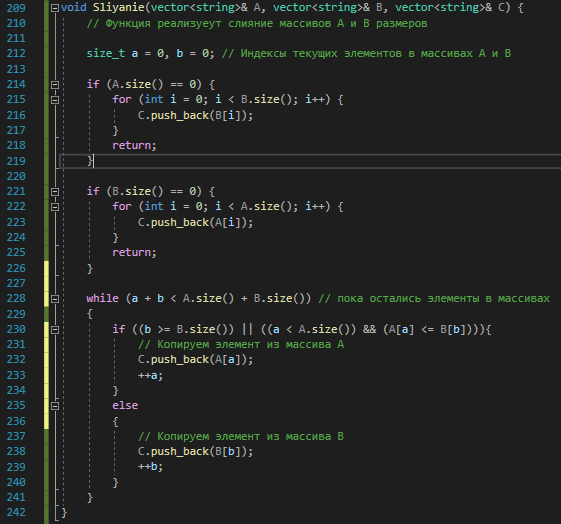


Рисунок 4 – Фаза слияния

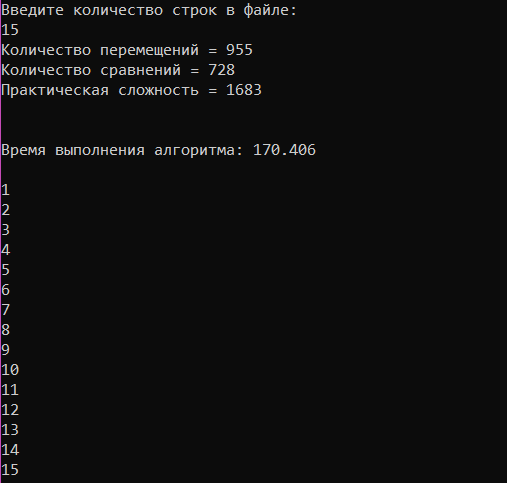
Протестируем алгоритм на примере, указанном выше:

Рисунок 5 – Тестирование алгоритма

## Тестирование алгоритма в соответствии с персональным вариантом

Файл с неотсортированными записями выглядит следующим образом:

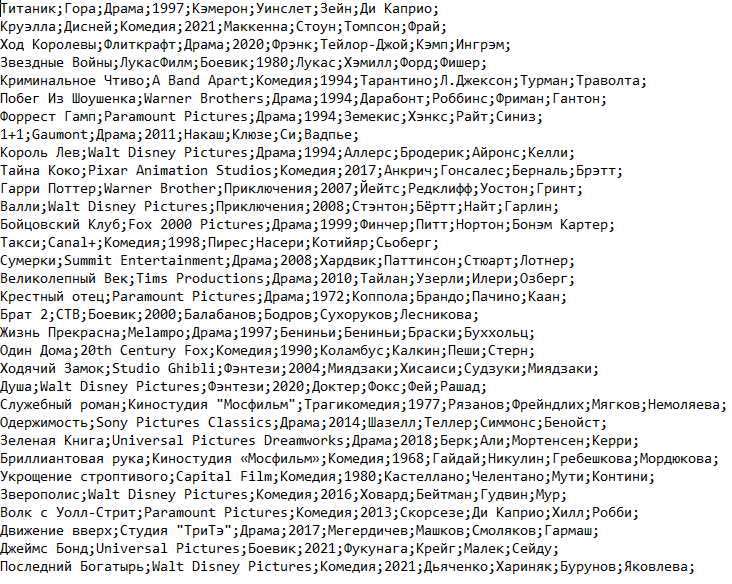


Рисунок 6 – Исходный файл данных

В соответствии с индивидуальным заданием требуется отсортировать данные по шестому столбцу – «Режиссёр».

Отсортированный файл:

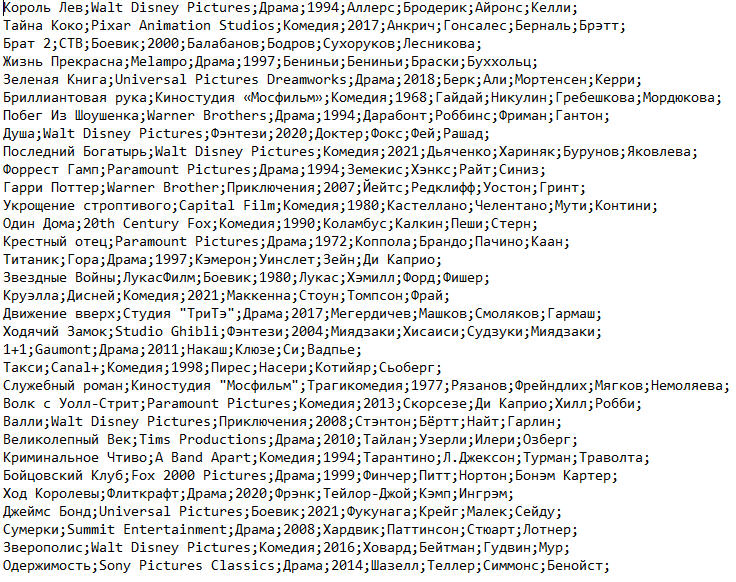


Рисунок 7 – Отсортированный файл

## Практическая сложность алгоритма

Проведя все нужные измерения, получаем следующее:

Таблица 1 – Таблица результатов

|  |  |  |
| --- | --- | --- |
| **Количество записей** | **T(n), мc** | **Тп(n)** |
| 8 | 129.9500 | 539 |
| 16 | 502.3310 | 2024 |
| 32 | 1099.9600 | 7865 |

# ЗАДАНИЕ №2

Разработать программу и применить алгоритм сортировки естественного слияния к сортировке файла с данными варианта

1. Постановка задачи

1. Реализовать функцию сортировки (возможно, с вспомогательными функциями) и основную подпрограмму main.
2. Отладить программу, протестировать на примере.
3. Адаптировать программу для сортировки файла с записями, протестировать на подготовленном ранее файле.
4. Сформировать таблицу результатов, указав количество записей и время сортировки.
5. Алгоритм сортировки естественного слияния

В алгоритме естественной сортировки учитывается изначальная упорядоченность значений. Для усовершенствования этой сортировки был предложен вариант предварительного разделения данных в файле на серии одной длины, загрузки каждой серии в оперативную память, сортировки этой серии другим алгоритмом и запись этих серий в исходный файл. Чем длиннее серию возможно выгрузить в память, отсортировать и вернуть в файл, тем эффективнее будет алгоритм самой сортировки.

Алгоритм использует тот же класс, что и предыдущая сортировка.

Далее следуют части кода с их описанием:

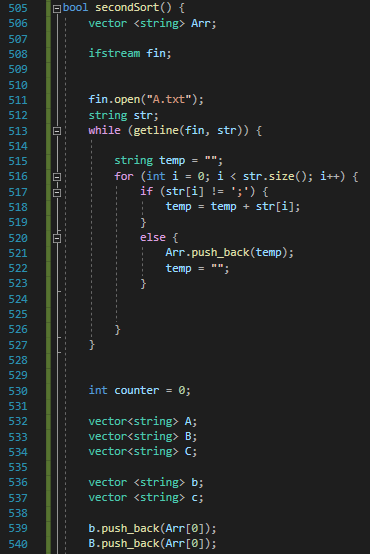


Рисунок 9 – Начало сортировки

Здесь происходит создание «буфера» для хранения серий из файла.

После этого начинается загрузка серий в массив буфера и их сортировка. После сортировки данные записываются в файлы В и С.

Дальше данные из файлов В и С сливаются в файл А

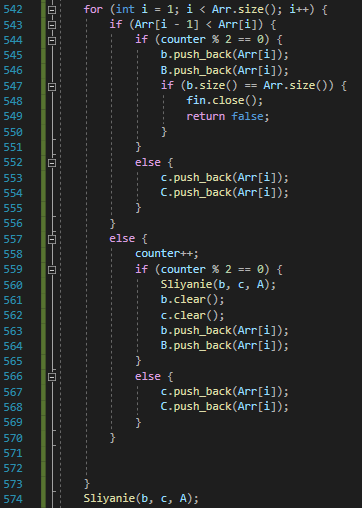


Рисунок 10 – Слияние отсортированных частей

Затем происходит фаза разделения – поочередное считывание данных в файлы В и С по размеру серии. Изначально серия равна размеру буфера.

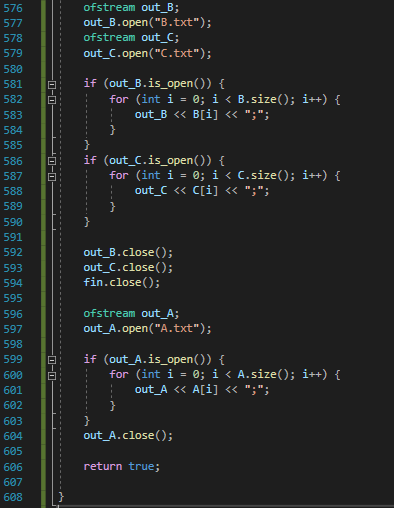


Рисунок 11 – Слияние с учетом ключей

После происходит слияние частей, учитывая ключевые значения. Так длина серии будет увеличиваться в два раза, пока не достигнет размера полной, изначальной серии размера n.

Протестируем алгоритм на примере:

Пусть есть файл А, содержащий записи с ключами: 17 31 5 59 13 41 43 67 11 23 29 47 3 7 71 2 19 57 37 61.

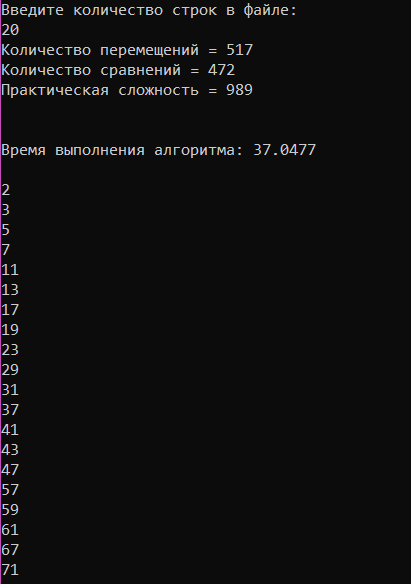
Итог сортировки:

Рисунок 14 – Тестирование алгоритма

1. Тестирование алгоритма в соответствии с персональным вариантом

В соответствии с индивидуальным заданием требуется отсортировать данные по шестому столбцу – «Режиссёр».

1. Практическая сложность алгоритма

Проведя все нужные измерения, получаем следующее:

Таблица 2 – Таблица результатов

|  |  |  |
| --- | --- | --- |
| **Количество записей** | **T(n), мc** | **Тп(n)** |
| 8 | 12.5429 | 305 |
| 16 | 39.671 | 774 |
| 32 | 49.1502 | 1724 |

# ВЫВОД

Была проделана большая работа, направленная на работу с алгоритмами внешней сортировки данных.

В ходе работы было установлено, что наиболее эффективным алгоритмом из исследуемых является сортировка естественным слиянием, она показала лучшие результаты, несмотря на то, что код программы был в два раза больше.