

Занятие 4.

Тема: «Алгоритмы внешних сортировок».

Цель: освоить приёмы сортировки данных из файлов.

1. Алгоритм сортировки прямого слияния для файлов

Фаза разделения:

1. Открыть файл А как входной.
2. Открыть файлы В и С как выходные (для записи).
3. Считываемые из А записи попеременно записываем в файлы В и С.
4. Закрываем файлы А, В, С.

Фаза слияния:

1. Открыть файл А как выходной (для записи).
2. Открываем файлы В и С как входные (для чтения).
3. Установить размер порции сливаемых данных: 1, 2, 4, 8 и т.д. для этого и следующих этапов.
4. Для каждой порции считываются по одной записи из файлов В и С.
5. Меньшая запись записывается в файл А, и считывается очередная запись из того файла, запись которого была переписана в файл А.
6. Пункты 4 и 5 повторяются до тех пор, пока записи очередной порции одного из файлов не будут исчерпаны.
7. Оставшиеся записи из порции другого файла переписываются в файл А.
8. Пункты с 4 по 7 повторяются до тех пор, пока не будет достигнут конец одного из файлов В и С. Тогда оставшиеся записи из другого файла переписываются в файл А.
9. Закрываются файлы А В С.

Сортировка завершается тогда, когда длина порции достигнет n .

2. Пример – демонстрация работы алгоритма прямого слияния на массиве ключей (в укороченном варианте)

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:

В: 8 13 15 9 3 5 1 14

С: 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

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.

3. Задание 1.

Разработать программу и применить алгоритм внешней сортировки **прямого слияния** к сортировке файла данных индивидуального варианта (приложение 1) по значению ключевого поля (ключ в структуре записи варианта – подчеркнутое поле).

- 1) Реализовать функцию сортировки (возможно, с вспомогательными функциями) и основную подпрограмму main.
- 2) Отладить программу, протестировать на примере из п.2.
- 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.

4. Алгоритм естественного слияния (пример работы на массиве ключей)

Сортировка *естественного* слияния, рассматривает две сливаемые подпоследовательности, как упорядоченные. Упорядоченные подпоследовательности принято называть *сериями*.

Пусть исходный файл разделен на два файла, каждый из которых содержит по n – серий (один может содержать $n-1$ серию). Тогда при слиянии этих файлов будет получен файл из n серий.

При каждом проходе число серий уменьшается вдвое, и общее число пересылок в худшем случае равно $n \log_2 n$, а в среднем меньше.

Процесс сортировки заканчивается, если при очередном проходе в файл будет перелита только одна серия.

Пример, для погружения в алгоритм сортировки естественным слиянием.

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

Выделим серии, завершая запятой, чтобы было нагляднее:

17 31' 5 59' 13 41 43 67' 11 23 29 47' 3 7 71' 2 19 57' 37 61

Получилось 7 серий.

Разделим файл на два файла В и С, переписывая в них поочередно по серии:

В: 17 31' 13 41 43 67' 3 7 71' 37 61

С: 5 59' 11 23 29 47' 2 19 57

Сольем файлы в файл А, сливая серии в упорядоченные серии

А: 5 17 31 59' 11 13 23 29 41 43 47 67' 2 3 7 19 57 71' 37 61

Опять разольем в В и С поочередно переписывая серии

В: 5 17 31 59' 2 3 7 19 57 71

С: 11 13 23 29 41 43 47 67' 37 61

Сливаем в файл А по сериям

А: 5 11 13 17 23 29 31 41 43 47 59 67' 2 3 7 19 37 57 61 71

Разливаем

и продолжаем до тех пор, пока в массив А не будет переписана серия длины n.

5. Алгоритм естественного слияния, оптимизированный для файлов

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, например, алгоритмом быстрой сортировки, 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.

Такое решение предлагается вам исследовать и разобраться в реализации. Рассмотрим алгоритм и его фазы. Он так же является двух фазным.

1. Определить размер свободной оперативной памяти для выгрузки в нее серии из файла. В программе создаем массив для хранения серии buf.
2. Открыть исходный файл А, подлежащий сортировке.
3. Открыть два файла для записи В и С.
4. Считать последовательность данных в количестве достаточном для размещения в массиве buf. Отсортировать в массиве методом внутренней сортировки и записать в файл В.
5. Считать следующую последовательность данных в количестве достаточном для размещения в массиве buf. Отсортировать в массиве методом внутренней сортировки и записать в файл С.
6. Пункты 4 и 5 выполнять, пока все данные из файла А не будут переписаны отсортированными во вспомогательные файлы В и С.
7. Слить данные в файл А сначала из файла В, затем из файла С. Теперь файл А содержит длинные упорядоченные серии, считаем, что данные в сериях упорядочены по возрастанию.
8. Фаза разделения включает поочередную запись серий из А в файлы В и С.

9. Фаза слияния имеет теперь следующий алгоритм:
- Считываем данные из одного и другого файлов, пока $a_i < a_{i+1}$, меньшее из сравниваемых записывать в файл A, пока одна из серий не будет исчерпана, тогда остаток другой переписываем в файл A, пока выполняется условие $a_i < a_{i+1}$.
 - После этого считываем следующую серию и так пока один из файлов не станет пустым, тогда серии другого переписываются в файл A.
10. Пункты 8 и 9 повторяются пока в файл A, в результате слияния не будет переписана только одна серия.

6. 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.

7. Формат отчета.

1. Для каждого алгоритма представить его словесное описание и блок-схему, код функций сортировки, результаты тестирования.
2. Отобразить 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.

Приложение 1. Индивидуальные варианты структуры записи файла.

№	Структура элемента
1	Сведения о студенте: <u>Номер зачетной книжки</u> , Фамилия, Имя, Дата рождения, Номер телефона, Дата поступления
2	Сведения о спортсмене: Фамилия, Имя, Дата рождения, Вес, <u>Рост</u> , Пол
3	Сведения о расписании занятий: Номер группы, название дисциплины, вид занятия (лек, лаб, практ), номер аудитории, <u>день недели</u> , номер пары,
4	Сведения о жителе: <u>Фамилия</u> , Город, Адрес: улица, дом, квартира.
5	Сведения о книге: Автор, Название, Инвентарный номер, <u>Издательство</u> , Количество страниц, Цена
6	Сведения об успеваемости одного студента по одной дисциплине: <u>Номер зачетной книжки</u> , Шифр группы, Название дисциплины, Дата получения оценки, Оценка, Фамилия преподавателя
7	Учет выдачи книг пользователям библиотеки. Карточка пользователя библиотеки содержит сведения, о выданной книге: <u>Номер читательского билета</u> , Инвентарный номер, Автор, Название, Дата выдачи, Дата возврата
8	Успеваемость группы студентов. Сведения по одному экзамену одного студента: Номер зачетной книжки, <u>Название дисциплины</u> , Дата получения оценки, Оценка, Фамилия преподавателя. (по каждому студенту будет столько записей сколько экзаменов он сдал). При вводе данных в строковые поля предусмотреть преобразование их к формату: первая буква большая, остальные маленькие
9	Список экспортируемых товаров. Об отдельном товаре хранятся данные: Наименование товара, <u>Страна импортирующая товар</u> , Количество(в штуках).
10	Магазин игрушек. Сведения об игрушке: <u>Название</u> (например: кукла, конструктор и т.д.), стоимость, возрастные границы детей (для кого игрушка предназначена) два поля – начальный возраст и конечный)
11	Служба знакомств. Структура данных хранит сведения о претендентах. Об отдельном претенденте: <u>Фамилия</u> , Имя Отчество, Возраст, Рост, Цвет глаз, Цвет волос, Зарплату, Наличие квартиры, Наличие машины.

12	Продажа квартир. Сведения о продаваемой квартире: <u>Общая площадь</u> , Жилая площадь, Площадь кухни, Наличие лоджии, Наличие санузла и его характеристики(совмещенный или нет), Район города
13	Поликлиника. Структура содержит данные о жителях, обслуживаемых данной поликлиникой. О жителе хранятся сведения: Фамилия , Имя, Отчество, Адрес, Место работы, <u>Дата прививки от гриппа</u>
14	ГИБДД. Структура хранит данные об автомашинах, О каждой машине: <u>Модель</u> , Номер(код региона, цифровой код, буквенный код), Цвет, Сведения о владельце(Фамилия, Имя , Адрес), дата последнего техосмотра.
15	Справочник владельца видеотеки. О каждом видеофильме хранятся данные: Название, Студия, Жанр, Год выпуска, <u>Режиссер</u> , Исполнители главных ролей(не более 10): фамилия.
16	Справочник фаната. Содержит данные о спортсменах: Анкетные и антропологические данные, Гражданство, <u>Вид спорта</u> , Клуб, Данные о личном рекорде(дата, призовое место)
17	Справочник туриста. Турагенство предлагает услуги: <u>Страна</u> , Город, Условия проживания (Отель-звезды, Автобус и т. д.), Условия проезда, Экскурсионное обслуживание, Сервис принимающей стороны, Стоимость путевки.