

10	Шейкерная сортировка	Быстрая сортировка (Хоара)
11	Шейкерная с условием Ай-версона	Быстрая сортировка (Хоара)
12	Сортировка Шелла со смещениями Д. Кнута. Способ 1	Быстрая сортировка (Хоара)
13	Шелла со смещениями Д. Кнута. Способ 2	Быстрая сортировка (Хоара)
14	Шелла со смещениями Р. Седжвика.	Быстрая сортировка (Хоара)
15	Пирамидальная сортировка	Быстрая сортировка (Хоара)
16	Турнирная сортировка	Быстрая сортировка (Хоара)

11.4. Практическая работа 4

Тема: «Алгоритмы внешних сортировок».

Задание 1. Разработать программу и применить алгоритм внешней сортировки *прямого слияния* к сортировке файла данных варианта по значению ключевого поля (табл. 8). Ключ в структуре записи варианта – подчеркнутое поле.

- 1) Файл данных варианта предварительно подготовить в тестовом файле с помощью любого тестового редактора.
- 2) Так как записи должны перемещаться, то удобнее хранить данные в двоичном файле, записи которого имеют фиксированную длину. Создать двоичный файл из записей, представленных в текстовом файле.
- 3) Разработать программу (функцию или несколько) сортировки.
- 4) Отладить программу, определить практическую сложность алгоритма для файлов с увеличивающимся количеством записей.
- 5) Сформировать таблицу результатов, указав количество записей и время сортировки.

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 выходкой (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 .

Задание 2. Разработать программу и применить алгоритм сортировки *естественного слияния* к сортировке файла с данными варианта (табл. 18).

- 1) Отладить программу, определить практическую сложность алгоритма для файлов с увеличивающимся количеством записей.
- 2) Сформировать таблицу результатов, указав количество записей и время сортировки.

Отчет:

1. Привести пример массива, заполненный случайными целыми числами. Провести его сортировку методом естественного слияния. Продемонстрировать процесс сортировки, представляя фазы разделения и слияния. Например, так, как указано в примере ниже.
2. Провести сортировку массива, который был использован в п.1 отчета, методом многофазного слияния. Продемонстрировать процесс сортировки, представляя фазы разделения и слияния. Например, так, как указано в примере ниже.
3. Представить отчет по разработке программы сортировки прямого слияния в соответствии с требованиями отчета по программе: постановка задачи, декомпозицию задачи (список алгоритмов, которые вы будет использовать при реализации алгоритма), код всей программы, результат тестирования (таблица с результатами).
4. Представить отчет по разработке программы сортировки естественного слияния в соответствии с требованиями отчета по программе: постановка задачи, де-

композицию задачи (список алгоритмов, которые вы будете использовать при реализации алгоритма), код всей программы, результат тестирования (таблица с результатами).

5. Выводы об эффективности алгоритмов на основе полученных практических замеров времени выполнения.

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.

Алгоритм естественного слияния для файлов

Для усовершенствования этой сортировки был предложен вариант, предварительного деления 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 *A*, подлежащий сортировке.
3. Открыть два файла для записи *B* и *C*.
4. Считать последовательность данных в количестве достаточном для размещения в массиве *buf*. Отсортировать в массиве методом внутренней сортировки и записать в файл *B*.
5. Считать следующую последовательность данных в количестве достаточном для размещения в массиве *buf*. Отсортировать в массиве методом внутренней сортировки и записать в файл *C*.
6. Пункты 4 и 5 выполнять, пока все данные из файла *A* не будут переписаны отсортированными во вспомогательные файлы *B* и *C*.

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 А, пока одна из серий не будет исчерпана, тогда остаток другой переписываем в файл А, 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 А.

10. Пункты 8 и 9 повторяются пока в файл А, в результате слияния не будет переписана только одна серия.

Таблица 18. 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.

11.5. Практическая работа 5

Тема: «Однонаправленный динамический список».

Задание. Реализуйте программу решения задачи варианта по использованию линейного однонаправленного списка.

Требования для всех вариантов:

1. Информационная часть узла определена вариантом (табл. 19).
2. Разработать функцию для создания исходного списка, используя функцию вставки нового узла перед первым узлом.
3. Разработать функцию вывода списка.
4. Разработать функции дополнительного задания варианта. При необходимости можно добавлять функции, декомпозируя задачу.
5. В основной программе выполните только тестирование каждой функции. Меню можно не создавать. Тесты обязательны.
6. Составить отчет по выполненному заданию.