

Курсовой проект
"Структуры и алгоритмы обработки данных"
для студентов факультета ИВТ
(варианты заданий)

Общая постановка задачи

Хранящуюся в файле базу данных (4000 записей) загрузить динамически в оперативную память компьютера в виде массива или списка (в зависимости от варианта), вывести на экран по 20 записей (строк) на странице с возможностью отказа от просмотра.

Упорядочить данные в соответствии с заданным условием упорядочения, используя указанный метод сортировки. Упорядоченные данные вывести на экран.

Предусмотреть возможность быстрого поиска по ключу в упорядоченной базе, в результате которого из записей с одинаковым ключом формируется очередь, содержимое очереди выводится на экран.

Из записей очереди построить дерево поиска по ключу, отличному от ключа сортировки, вывести на экран содержимое дерева и предусмотреть возможность поиска в дереве по запросу.

Закодировать файл базы данных статическим кодом, предварительно оценив вероятности всех встречающихся в ней символов. Построенный код вывести на экран, вычислить среднюю длину кодового слова и сравнить ее с энтропией исходного файла.

Дополнительное задание. Упакованную базу данных записать в файл, вычислить коэффициент сжатия данных.

При выполнении задания главное внимание следует уделить эффективности применяемых алгоритмов, исключению всех лишних операций, а также эффективному использованию динамической памяти, исключению дублирования данных при распределении в памяти. Операции, выражающие логически завершенные действия, рекомендуется оформлять в виде подпрограмм, грамотно выбирая между процедурами и функциями. Имена переменных и подпрограмм, параметры подпрограмм, используемые языковые конструкции должны способствовать удобочитаемости программы.

Для сравнения символьных строк не рекомендуется пользоваться встроенными языковыми средствами и библиотечными функциями.

В целях способствования более глубокому пониманию авторами своих программ, последние не должны содержать поясняющие комментарии (здесь мы вынуждены отойти от хорошего стиля программирования).

Вариант задания задается с помощью чисел В, С, S, D, E, где

В - номер базы данных;

С - вариант условия упорядочения для этой базы данных;

S - метод сортировки;

D - тип дерева поиска;
E - метод кодирования.

Условие упорядочения, как правило, представляет собой сложный ключ, состоящий из двух полей (первое поле – старшая часть ключа, второе поле – младшая). Упорядочение данных необходимо произвести за один проход сортировки. Ключ поиска указывается вместе с условием упорядочения и, как правило, представляет собой упрощенный вариант ключа сортировки.

Числа B, C, S, D, E определяются с помощью таблицы соответствия вариантов, приведенной ниже. Каждый студент разрабатывает программу для одного варианта, номер варианта выдается преподавателем. Допускаются различные творческие дополнения, ведущие в сторону развития. Выполнение работы по чужому варианту не допускается.

В ходе работы должна быть создана программа, выполняющая поставленную задачу, и оформлен отчет, включающий в себя следующие разделы:

Титульный лист

Содержание

1. Постановка задачи (для конкретного варианта)

2. Основные идеи и характеристики применяемых методов

2.1. Метод сортировки

2.2. Двоичный поиск

2.3. Списки и очереди

2.4. Вид дерева и поиск

2.5. Метод кодирования

3. Описание структур данных и использованных алгоритмов

3.1. Используемые структуры данных

3.2. Особенности реализации алгоритмов

4. Описание программы

4.1. Основные переменные и структуры

4.2. Описание подпрограмм

(название, параметры, что делает)

5. Исходный текст программы

(выделить заголовки подпрограмм)

6. Результаты

(показательные фрагменты, 3-5 страниц)

7. Выводы

(как решена поставленная задача)

Шрифт отчета произвольный, размер шрифта - 12.

К услугам студентов предоставляются файлы баз данных для тестирования программ (имена файлов указаны далее в тексте) и программа просмотра баз данных. Все файлы доступны в каталоге

Варианты баз данных (БД)

Общие замечания

1. Все текстовые поля следует рассматривать как символьные массивы (array of char), а не строки (string). Это сделано для совместимости между языками программирования, а также из-за того, что в базах данных не принято хранить лишнюю информацию, такую как длина строки. Если длина поля превышает размер хранимой в нем информации, то оно дополняется пробелами справа. Каждое текстовое поле имеет свой формат, который определяет смысл записанных в него данных.

При описании формата в угловых скобках < > указываются отдельные его элементы (сами угловые скобки в состав текста не входят); пробелы обозначаются с помощью символа подчеркивания. Если поле включает только один текстовый элемент, то формат не указывается.

2. Целочисленные поля представляются 16-разрядными положительными числами.

3. При описании структуры записей в программах необходимо точно соблюдать порядок и размер полей.

4. Для числовых полей использовать тип short int, а для базы номер 3 использовать unsigned short int.

B = 1 (файл testBase1.dat)

Библиографическая база данных "Жизнь замечательных людей"

Структура записи:

Автор: текстовое поле 12 символов

формат <Фамилия>_<буква>_<буква>

Заглавие: текстовое поле 32 символа

формат <Имя>_<Отчество>_<Фамилия>

Издательство: текстовое поле 16 символов

Год издания: целое число

Кол-во страниц: целое число

Пример записи из БД:

Кловский В Б

Лев Николаевич Толстой

Молодая гвардия

1963

864

Варианты условий упорядочения и ключи поиска (K):

S = 1 - по фамилиям(!) замечательных (!) людей, K = три первые

буквы фамилии;

C = 2 - по году издания и автору, K = год издания;

C = 3 - по издательству и автору, K = три первые буквы издательства.

B = 2 (файл testBase2.dat)

База данных "Предприятие"

Структура записи:

ФИО сотрудника: текстовое поле 30 символа

формат <Фамилия>_<Имя>_<Отчество>

Номер отдела: целое число

Должность: текстовое поле 22 символа

Дата рождения: текстовое поле 10 символов

формат дд-мм-гг

Пример записи из БД:

Петров_Иван_Иванович_____

130

начальник_отдела_____

15-03-46

Варианты условий упорядочения и ключи поиска (K):

C = 1 - по номеру отдела и ФИО, K = номер отдела;

C = 2 - по дням(!) рождения и ФИО, K = день рождения;

C = 3 - по дате(!) рождения, K = год рождения.

B = 3 (файл testBase3.dat)

База данных "Обманутые вкладчики"

Структура записи:

ФИО вкладчика: текстовое поле 30 символа

формат <Фамилия>_<Имя>_<Отчество>

Сумма вклада: целое число (использовать unsigned short int)

Дата вклада: текстовое поле 10 символов

формат дд-мм-гг

ФИО адвоката: текстовое поле 22 символа

формат <Фамилия>_<буква>_<буква>

Пример записи из БД:

Петров_Иван_Федорович_____

130

15-03-46

Иванова_И_В_____

Варианты условий упорядочения и ключи поиска (K):

C = 1 - по ФИО и сумме вклада, K = первые три буквы фамилии;

C = 2 - по сумме вклада и дате, K = сумма вклада;

C = 3 - по ФИО адвоката и сумме вклада, K = первые три буквы фамилии адвоката.

V = 4 (файл testBase4.dat)

База данных "Населенный пункт"

Структура записи:

ФИО гражданина: текстовое поле 32 символа

формат <Фамилия>_<Имя>_<Отчество>

Название улицы: текстовое поле 18 символов

Номер дома: целое число

Номер квартиры: целое число

Дата поселения: текстовое поле 10 символов

формат дд-мм-гг

Пример записи из БД:

Петров_Иван_Федорович_____

Ленина_____

10

67

29-02-65

Варианты условий упорядочения и ключи поиска (K):

C = 1 - по ФИО и названию улицы, K = первые три буквы фамилии;

C = 2 - по названию улицы и номеру дома, K = первые три буквы названия улицы;

C = 3 - по дате поселения и названию улицы, K = год поселения.

Варианты методов сортировки

S = 1 Метод Вилльямса-Флойда

Файл базы данных загружается в динамическую память с формированием индексного массива как массива указателей.

S = 2 Метод Хоара

Файл базы данных загружается в динамическую память с формированием индексного массива как массива указателей.

S = 3 Метод прямого слияния

Файл базы данных загружается в динамическую память в виде списка, сортировка проводится с использованием очередей, затем, для проведения быстрого поиска, по отсортированному списку строится индексный массив.

S = 4 Цифровая сортировка

В качестве ключа для упорядочения нужно взять всего по несколько (обычно не менее трех) байт из соответствующих полей. Файл базы данных загружается в динамическую память в виде списка, который сортируется цифровым методом. Затем, для проведения быстрого поиска, по отсортированному списку строится индексный массив.

Типы деревьев поиска

D = 1 AVL-дерево

D = 2 Двоичное B-дерево

D = 3 Дерево оптимального поиска (приближенный алгоритм A1)

D = 4 Дерево оптимального поиска (приближенный алгоритм A2)

Методы кодирования

E = 1 Код Хаффмена

E = 2 Код Шеннона

E = 3 Код Фано

E = 4 Код Гилберта-Мура

Таблица соответствия вариантов доступна в файле
Таблица_вариантов.docx