

## 2003 年山东大学数据结构笔记

考研加油站收集整理 <http://www.kaoyan.com>



2003《数据结构》 2003.12.30.

1. 题型有变化: 简答 + 计算 (元项等等) 大约 3:2

## CH1 绪论.

概念: 1.1 理解. 什么是“数据结构”是——

1.2 基本概念和术语.

数据、数据项、数据类型及相关概念 (图体信)

1.3 抽象数据类型表示与实现.

尽量用 (语言) 描述.

1.4 算法与算法分析: 时间/空间复杂度. 求解时间复杂度 (计算)  $O(f(n))$ 

语句频度

CH2 线性表 (概念不多, 主要是算法设计!)

1. 结构: 顺序表与链表 (单、双、单循环、双循环) 区别, 具体结构, 特点.

2. 线性表定义: 1. 概念

2. 应用.

/ 不要直接调用, 要具体描述一下, 基本操作 + /

2.2 线性表顺序表示和实现:

① 顺序存取特点, 运算规律.

② 算法设计: 时间复杂度分析 (先设计结构, 再分析时间复杂度, 有: 时间、空间限制)

掌握书上内容学会应用, 一般存储结构确定, 要求实现达到——

2.3 线性表链式表示和实现.

① 单链表, 双链表, 单循环链表, 双循环链表 同样多行, 重视 (结构, 特点)

② 算法设计: 时间复杂度分析

2.4 一元多项式表示和相加: 用单链表表示, 要求会, 利用单链表操作完成多项式 (一元) 的运算.

CH3 栈和队列: 结构, 栈, 队列 → 一般, 循环, 双.

3.4 一般 先进先出, 双端 两端均可出入

输入/输出受限的双端队列

输入受限, 输出受限

队列: 链队列, 循环队列 → 上道, 假溢出.

3.1 栈 链栈 (栈)

顺序, 栈底

先进后出

结构: 例: 给输入 1, 2, 3, 4, 求可能输出的结果. (用栈, 不用队列)

应用: 递归活用

1. 栈队列上的基本操作 (入栈, 出栈, 判空, 取头点, 入队, 出队, 判空, 求元素个数).  
基本操作的应用 (在后面体现在树, 图第算法中用栈时可以直接引用关于栈队列的基本操作).

2. 基本知识: 队列元素个数? 判满条件? 判空条件? 判满判断? 链式, 顺序? 不空, 判空, 判满, 判空.

3. 栈的应用举例.

3.2.1 3.2.2 3.2.3 不一定用栈. / 很有可能不用.

3.3 栈与递归的实现.

a 问题 → 递归定义 → 递归算法的设计

b 栈在递归中的作用, 实现递归的原理及方法.

栈如何实现递归? 执行递归时栈的变化过程要求掌握.

c 递归到非递归的转化 (限定用栈).

CH4 串

4.1 串的基本操作的含义及其应用.

4.2 串的存储和实现.

与线性表 (顺序, 链式) 相结合实现串上指定的操作 (算法设计).

4.3 串的模式匹配算法: 改进的 KMP. (熟练掌握) 思想, 方法, 模式匹配的存储.

例: P82, 图 4.5.

next[j] 和 nextval[j] 完全由模式串决定.

例:  $next[j] = \begin{cases} 0 & j=1 \\ max\{k | 1 \leq k < j \text{ 且 } s_1 \dots s_k = s_{j-k+1} \dots s_j \} & j > 1 \end{cases}$

4.4 看开环思路.

CH5 数组和广义表.

5.1 一般数组的概念.

5.2 数组的顺序表示.

5.3 特殊矩阵.

稀疏矩阵: 存储方式 = 三元组顺序表, 行逻辑链接顺序表, + 十字链表法.

各种存储表的特征及实现指定运算的方法.

算法设计: 二维数组上算法设计.

2



5.4 二叉树的基本概念

取表头、表尾、广义表上的运算 (Get head, Get tail) P108.

5.5 二叉树的存储结构

会画：画前说叶下

CH6 树和二叉树

6.1 树的定义和基本术语 (概念的理解和应用)

6.2 二叉树 (重要)

结构、性质及其应用、存储结构

满二叉树

完全二叉树

6.3 遍历、线索二叉树

遍历 { 操作的应用, 写序列, 由序列构造二叉树  
算法: 递归  $\leftrightarrow$  非递归 { 用栈  
不用栈

线索化、前、中、后序

前继后继

二叉树 (线索二叉树) 上算法设计 { 递归  
非递归

二叉树的应用, 树的变化过程

6.4 树和森林

存储表示

双亲

孩子

孩子兄弟 (左兄弟)

森林、树、二叉树的转换

对森林 遍历序列

6.6 赫夫曼树及其应用

赫夫曼树的概念: 带权路径长度最小 (wpl)

应用: 由作为编码  $\rightarrow$  构造赫夫曼树

6.8 中序遍历

图 6.33. 其中一些构造二叉树的方法

二叉树的计数, 二叉树构造二叉树, 前序  $\rightarrow$  中序

12月1日 下午2:00

## 第十二章 文件

12.4, 12.5 不考。

什么是文件、什么是索引、多重表文件、倒排文件区别。

## 第七章 图 (十字链表 7.2.3 + 7.4.2 + 7.4.4 不考) (习题集要做) (算法较少)

1. 图的定义及基本特征: 有向、无向图, 连通、强连通图 (黑体字) 完全图。

顶点数和边数关系 (元素)。

2. 图的存储结构: 邻接矩阵, 邻接表, 多重邻接表。

3. DFS, BFS: 思想、实例 (算法) 画图后写比较稳妥。

4. 生成树、最小生成树 (概念)。

概念、算法的思想: 普里姆算法, 克鲁斯卡尔算法 (不用写具体算法, 及构造实例)。  
DFS, BFS 生成树。

5. AOV, AOE 网络, 概念, 应用。

AOV 用于拓扑排序的实例和 (算法), AOE 用于工程问题的实例。

6. 最短路径 (单源点到其他顶点, 所有顶点对)。  
每个事件的最晚、早发生时间。  
工程关键路径, 关键活动。

单源最短, 求解过程实例。

所有对, 推导方法, 过程实例。

## 第九章 (9.1.3, 9.2.3 不考)。

1. 静态表 (思想、效率分析)。

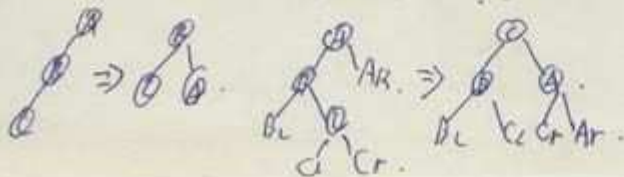
顺序查找。

有序表的折半查找 (递归实现)。

静态链表不考。

索引顺序表的分块查找:  $(n = b \text{ 块} \cdot s \text{ 个/块})$  求  $b$  和  $s$  的最小值。

2. 动态表:

(1) B+ 排序树及相关 (算法) 插入、删除 (叶子结点, 或非叶子, 或 B+ 树) 判相等。  
判定所在层次。(2) AVL 树: 概念 (平衡因子  $BF$ )、调整原理、方法、规则、实例。

高度和结点数的关系。

4

1. B-树 B+树 (是在B-树上)

概念 (区别) 阶数  $m$  和节点数、key 数关系 ( $m=3$  时, 叫 2-3 树)  
高度  $h$ .

B-树的插入、删除、后的调整 (节点)

9.3 HASH 表:

组织思想: 常用的 HASH 函数和冲突处理技术

对实例构造 Hash 表, 并分析成功比较次数。

P261-P262 必考。

第十章

每种排序法, 思想, 实例, 时间/空间复杂度分析,  $P269$ , (平均, 最好, 最坏) 稳定性。

1. 直接插入 2. SHELL 3. 简单选择 4. 快速 5. 堆 6. 归并 7. 基数

8. 起泡

算法掌握: 2. 希尔, 4. 快速 5. 堆 7. 基数

递归 非递归

补充:

1. B-树, 10 阶, key 的数目  $[1, 3]$ , 节点数



2. 判断一棵二叉树是否是排序树 (2002 年有试题): 如 5-中序遍历

3. P153 图 2.29 求关键路径。

4. P157 图 2.34 求 10 到其他点的最短路径 (无  $V_1$ )。

$$5. A^0 = \text{cost} = \begin{pmatrix} 0 & 4 & 11 \\ 6 & 0 & 2 \\ 3 & \infty & 0 \end{pmatrix} \quad A^1 = \begin{pmatrix} 0 & 4 & 11 \\ 6 & 0 & 2 \\ 3 & 7 & 0 \end{pmatrix} \quad A^2 = \begin{pmatrix} 0 & 4 & 6 \\ 6 & 0 & 2 \\ 3 & 7 & 0 \end{pmatrix} \quad A^3 = \begin{pmatrix} 0 & 4 & 6 \\ 5 & 0 & 2 \\ 3 & 7 & 0 \end{pmatrix}$$

6. 树论: 树的重要, 链表有, 图减少。

7.  $n$  个顶点的有向强连通图最少有几条边?  $n$ 。

8. 10 个顶点的  $n$  阶图至少有一个顶点, 最少有一个顶点。

9. 归并排序所需额外空间  $m+n$  ( $m, n$  为归并元素个数)。

10. 两个序列比较最少需  $\frac{1}{2}n \log n$  次, 最多需  $n \log n$  次。

5