

备考计算机专业研究生考试通用教材

文都教育

全国硕士研究生入学统考 计算机专业基础综合 要点速记手册

10NIANQUANGUOSHUSHIYANJIUSHENGRUXUETONGKAOKAOYANJISUANJIZHUANYEJICHUZONGHESUJISHOUCE

策划◎文都考研命题研究中心
主编◎崔巍 于晖

概念·考点

轻松记忆



原子能出版社

备考计算机专业研究生考试通用教材

 文都教育

全国硕士研究生入学统考

计算机专业基础综合

要点速记手册

 原子能出版社

图书在版编目(CIP)数据

全国硕士研究生入学统考计算机专业基础综合要点速记手册/崔巍,于晖编著. —北京:原子能出版社,2008.12
ISBN 978-7-5022-4441-5

I. 全… II. ①崔…②于… III. 电子计算机—研究生—入学考试—自学参考资料 IV. TP3

中国版本图书馆 CIP 数据核字(2008)第 208271 号

全国硕士研究生入学统考计算机专业基础综合要点速记手册

出版发行:原子能出版社(北京市海淀区阜成路 43 号 100048)

责任编辑:谭 俊

特约编辑:刘 燕 师 潭

封面设计:刘艳南

印 刷:北京九天志诚印刷有限公司

经 销:全国新华书店

开 本:880×1230 毫米 1/64

印 张:4.25

字 数:130 千字

版 次:2009 年 2 月第 1 版 2009 年 2 月第 1 次印刷

书 号:ISBN 978-7-5022-4441-5

定 价:10.00 元

版权所有 侵权必究 网址:<http://www.aep.com.cn>

前 言

- ◆ 结构图解 全面精炼大纲
- ◆ 重点突出 浓缩考试要点
- ◆ 难点释疑 提高应试能力

计算机学科专业基础综合科目包括数据结构、计算机组成原理、操作系统、计算机网络等四门课程,内容繁多,涉及面广,在复习中很多同学感觉到花费时间颇多但收获却不太理想。为了帮助广大考研学生方便学习、轻松记忆,根据《全国硕士研究生入学统一考试计算机学科专业基础综合考试大纲》规定的考试内容和考试要求,结合学生实际需求,编写此速记手册。本手册内容全面,每章由以下内容构成:

第一部分,“知识结构图”,高度概括了考试大纲,能够提纲挈领地把握每章主要内容以及各知识点之间的逻辑关系,更为学生对于每部分知识的掌握给出了记忆线索。

第二部分,“重点归纳”,对大纲知识点进行准确、完整的阐释,条理清晰,知识点明确,用语精炼,便于理解与记忆,使考生复习能够真正地做到有的放矢,目标明确。

第三部分,“难点释疑”,对于考生在复习过程的难点以及易混、易错的部分进行答疑解惑,避免考生在考试过程中的不必要失分,达到节约备考时间,提高考试成绩的目的。

由于作者水平有限,时间也比较仓促,尽管经过反复校对与修改,书中难免还存在错漏和不妥之处,敬请广大读者和专家批评指正。

衷心地希望本手册能帮助考生在考试中取得理想的好成绩!

作者

2009年2月

目 录

第一部分 数据结构	(1)
第1章 绪论	(2)
第2章 线性表	(8)
第3章 栈、队列和数组	(22)
第4章 树和二叉树	(34)
第5章 图	(57)
第6章 查找	(69)
第7章 排序	(82)
第二部分 计算机组成原理	(87)
第1章 计算机系统概述	(88)
第2章 数据的表示和运算	(94)
第3章 存储器层次结构	(110)
第4章 指令系统	(123)
第5章 中央处理器(CPU)	(129)
第6章 总线	(146)
第7章 输入输出(I/O)系统	(150)

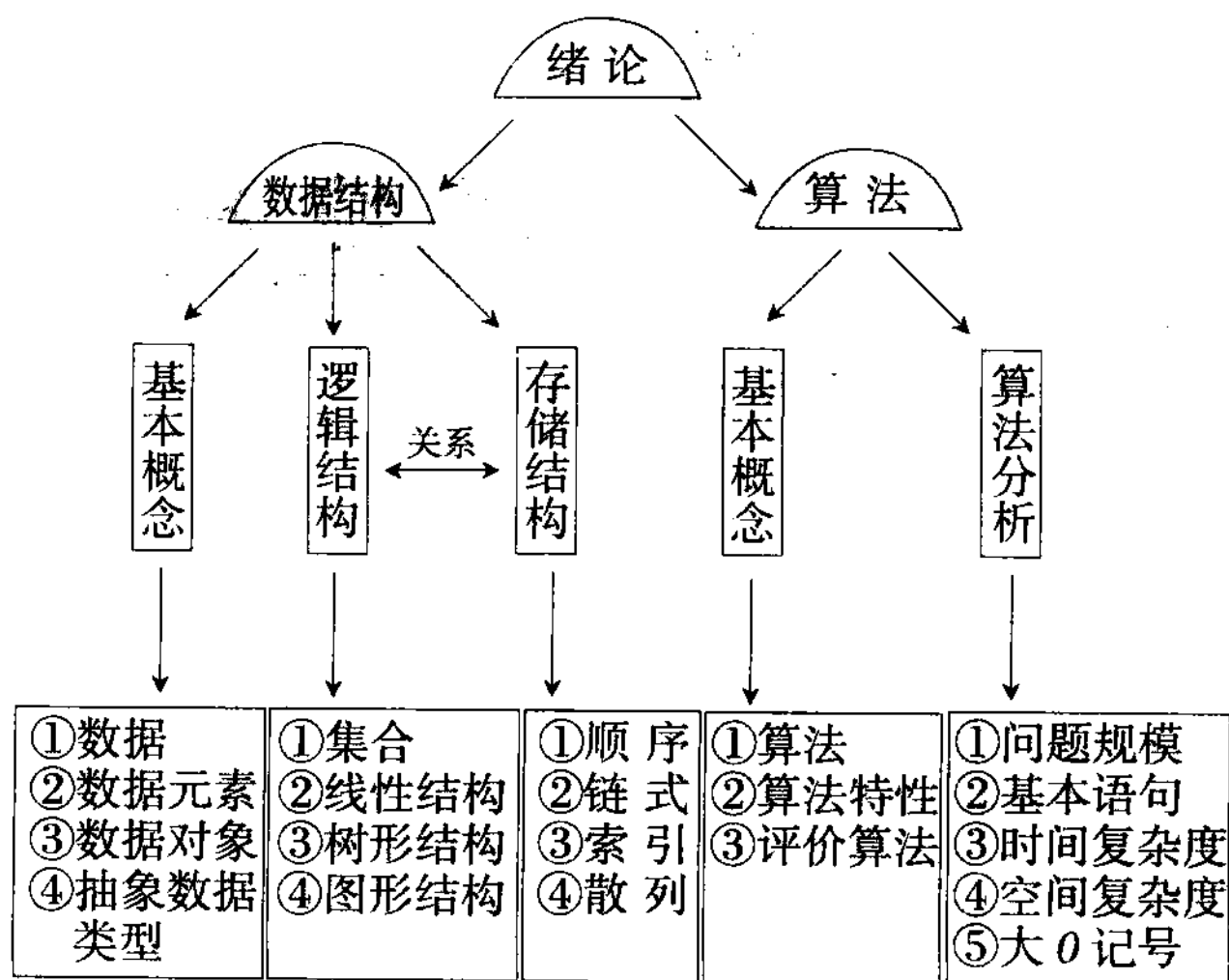
第三部分 操作系统	(155)
第1章 操作系统概述	(156)
第2章 进程管理	(162)
第3章 内存管理	(186)
第4章 文件管理	(198)
第5章 输入输出(I/O)管理	(205)
第四部分 计算机网络	(215)
第1章 计算机网络体系结构	(216)
第2章 物理层	(219)
第3章 数据链路层	(223)
第4章 网络层	(235)
第5章 传输层	(245)
第6章 应用层	(252)

第一部分

数 据 结 构

第 1 章 绪论

一、知识结构图





二、重点归纳

1. (1) **数据结构**课程主要是研究非数值计算的程序设计问题中的计算机操作对象以及它们之间的关系和操作的学科。

(2) **数据结构**是指互相之间存在着一种或多种关系的数据元素的集合。

(3) **数据结构**是一个二元组 $\text{Data_Structure} = (D, R)$, 其中, D 是数据元素的有限集, R 是 D 上关系的有限集。

2. **逻辑结构**是指数据之间的相互关系。通常分为四类结构:

(1) **集合**: 结构中的数据元素除了同属于一种类型外, 别无其它关系。

(2) **线性结构**: 结构中的数据元素之间存在一对一的关系。

(3) **树型结构**: 结构中的数据元素之间存在一对多的关系。

(4) **图状结构或网状结构**: 结构中的数据元素之间存在多对多的关系。

3. **存储结构**是指数据结构在计算机中的表示, 又称为数据的物理结构。通常由四种基本的存储方法实现:

(1)顺序存储方式。数据元素顺序存放,每个存储结点只含一个元素,存储位置反映数据元素间的逻辑关系。存储密度大,有些操作(如插入、删除)效率较差。

(2)链式存储方式。每个存储结点除包含数据元素信息外还包含一组(至少一个)指针,指针反映数据元素间的逻辑关系。这种方式不要求存储空间连续,便于动态操作(如插入、删除等),但存储空间开销大(用于指针),另外不能折半查找。

(3)索引存储方式。除数据元素存储在一组地址连续的内存空间外,还需建立一个索引表,索引表中索引指示存储结点的存储位置(下标)或存储区间端点(下标)。

(4)散列存储方式。通过散列函数和解决冲突的方法,将关键字散列在连续的有限的地址空间内,并将散列函数的值解释成关键字所在元素的存储地址。其特点是存取速度快,只能按关键字随机存取,不能顺序存取,也不能折半存取。

4. 算法是对特定问题求解步骤的一种描述,是指令的有限序列。其中每一条指令表示一个或多个操作。

特性:①有穷性;②确定性;③可行性;④输入;⑤输出。

评价算法的几个方面:①正确;②可读;③健壮;④高效。

5. 时间复杂度:以基本运算的原操作重复执行的次数作为算法的时间度量。一般情况下,算法中原操作重复执行的次数是规模 n 的某个函数 $f(n)$ 。

常见的渐进时间复杂度有:

$$O(1) < O(\log_2 n) < O(n) < O(n \log_2 n) < O(n^2) < O(n^3) < O(2^n) < O(n!) < O(n^n)$$

注意:有的情况下,算法中基本操作重复执行的次数还随问题的输入数据集不同而不同。

6. 算法的存储空间度量:若输入数据所占空间只取决与问题本身,和算法无关,则只需要分析除输入和程序之外的辅助变量所占额外空间。

原地工作:若所需额外空间相对于输入数据量来说是常数,则称此算法为原地工作。

7. 熟悉以下程序段的语句频度:

```
(1) for (int i = 1; i <= n; i++)  
    for (int j = 1; j <= i; j++)
```

S;

S 的语句频度是 $n(n+1)/2$ 。

(2) for(i = 1; i <= n; i++)

for (j = 1; j <= i; j++)

for (k = 1; k <= j; k++)

S;

S 的语句频度是 $n(n+1)(n+2)/6$ 。



三、难点释疑

1. 关于逻辑结构,以下几点需特别注意:

(1) 逻辑结构与数据元素本身的形式、内容无关。

(2) 逻辑结构与数据元素的相对位置无关。

(3) 逻辑结构与所含数据元素的个数无关。

(4) 逻辑结构与数据的存储无关,它是独立于计算机的。

2. 数据的逻辑结构和存储结构是密切相关的两个方面。一般地,一种数据的逻辑结构根据需要可用多种存储结构来存储,而采用不同的存储结构,其数据处理的效率往往是不同的。

3. 算法与程序:十分相似,但又有区别。

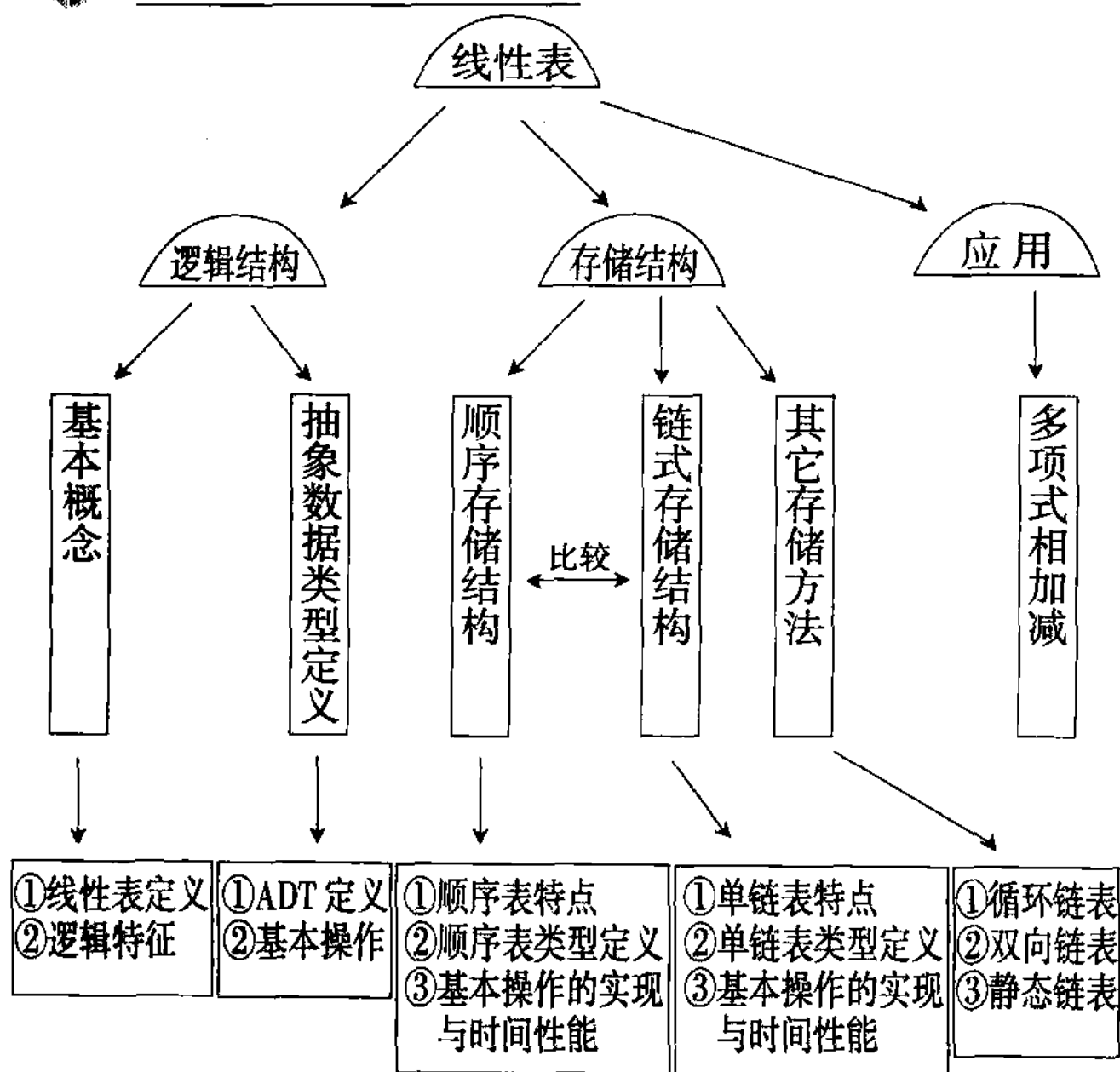
一个程序不一定满足有穷性。程序中的指令必

须是机器可执行的,而算法中的指令则无此限制。算法代表了对问题的解,而程序则是算法在计算机上的特定的实现。一个算法若用程序设计语言来描述,则它就是一个程序。

4. **算法与数据结构**:是相辅相承的。解决某一特定类型问题的算法可以选定不同的数据结构,而且选择恰当与否直接影响算法的效率。反之,一种数据结构的优劣由各种算法的执行来体现。

第2章 线性表

一、知识结构图





二、重点归纳

1. 线性表是具有相同数据类型的 n ($n \geq 0$) 个数据元素的有限序列,是最简单、最基本、也是最常用的一种线性结构。

2. 顺序表的形式描述:

静态分配: #define LISTSIZE 100

```
typedef struct {
    ElemType elem[ LISTSIZE ];
    int length;
} Sqlist;
```

动态分配: #define LIST_INIT_SIZE 100

#define LISTINCREMENT 10

```
typedef struct {
    ElemType * elem;
    int length;
    int listsize;
} Sqlist;
```

顺序表上基本运算的实现

(1) 顺序表具有按数据元素的序号随机存取的特点,时间复杂度为 $O(1)$ 。

(2) 按值 x 查找: 主要运算是比较, 比较的次数与

值 x 在表中的位置有关,也与表长有关,平均比较次数为 $(n+1)/2$,时间复杂度为 $O(n)$ 。

(3) **插入运算**:在第 i 个位置上插入 x ,从 a_n 到 a_i 都要向下移动一个位置,共移动 $n-i+1$ 个元素。

等概率情况下,平均移动数据元素的次数:

$$E_{in} = \sum_{i=1}^{n+1} p_i (n-i+1) = \frac{1}{n+1} \sum_{i=1}^{n+1} (n-i+1) = \frac{n}{2}$$

说明:需移动表中一半的元素,时间复杂度为 $O(n)$ 。

(4) **删除运算**:删除第 i 个元素,从 a_{i+1} 到 a_n 都要向上移动一个位置,共移动 $n-i$ 个元素。

等概率情况下,平均移动数据元素的次数:

$$E_{de} = \sum_{i=1}^n p_i (n-i) = \frac{1}{n} \sum_{i=1}^{n+1} (n-i) = \frac{n-1}{2}$$

说明:大约需要移动表中一半的元素,时间复杂度为 $O(n)$ 。

3. 单链表形式描述:

```
typedef struct LNode {  
    ElemType data;  
    struct LNode * next;  
} LNode, * LinkList;  
LinkList Head;
```

单链表上基本运算的实现

(1) 建立带头结点的单链表