

全国计算机等级考试(新大纲)应试指导

二级

公共基础知识

考题精解与 考场模拟

■ 蔡学望 田昆 刘金 编著



紧扣最新考纲 预测出题趋势
分析考题思路 指明解题方法
传授应试经验 模拟考场试题

全国计算机等级考试(新大纲)应试指导

二级

公共基础知识

考题精解与 考场模拟

■ 蔡学望 田昆 刘金 编著



人民邮电出版社

图书在版编目 (CIP) 数据

二级公共基础知识考题精解与考场模拟 / 蔡学望, 田昆, 刘金编著.

—北京: 人民邮电出版社, 2005.4

(全国计算机等级考试(新大纲)应试指导)

ISBN 7-115-13258-5

I. 二... II. ①蔡...②田...③刘... III. 电子计算机—水平考试—自学参考资料

IV. TP3

中国版本图书馆 CIP 数据核字 (2005) 第 026033 号

内 容 提 要

全书按照教育部考试中心最新修订的《全国计算机等级考试考试大纲 (2004 年版)》中对二级考试公共基础知识部分的要求而编写。

全书主要内容包括: 数据结构与算法、程序设计基础、软件工程基础以及数据库设计基础等基本知识、出题要点、出题趋势和应试指导。在每一章后结合知识点设计并分析了大量模拟试题和实战习题。在第 5 章中设计了 10 套笔试模拟试卷, 并对答案进行了分析。

本书内容紧扣考试大纲, 对参加全国计算机等级考试二级公共基础知识应试者, 是必备的考前应试辅导教材。

全国计算机等级考试 (新大纲) 应试指导 二级公共基础知识考题精解与考场模拟

◆ 编 著 蔡学望 田 昆 刘 金

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

读者热线 010-67170985

河北涞水华艺印刷厂印刷

新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 12.5

字数: 298 千字

2005 年 4 月第 1 版

印数: 1—5 000 册

2005 年 4 月河北第 1 次印刷

ISBN 7-115-13258-5/TP · 4568

定价: 17.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

全国计算机等级考试应试指导丛书编委会

顾 问 史济民 中国计算机基础教育研究会荣誉副理事长

李志蜀 教育部计算机基础教学课程指导委员会委员

丛书主编 周启海

编 委 (排名不分先后)

杜小丹 蔡学望 张钟澍 陈杰华 鄢 涛

李 梅 叶 斌 陈高云 董继菡 黄晓伶

白晓毅 向 伟 滑 玉

编者的话

本书是根据教育部考试中心制定的《全国计算机等级考试考试大纲（2004 年版）》中对二级计算机公共基础知识的要求，并紧扣新大纲公共基础知识的基本出题点，结合近几年计算机等级考试的新情况、新思路进行编写的。

根据 2004 年版新的大纲要求，对于二级考试中的所有科目增加基础知识部分，并且对基础知识统一要求，使用统一的基础知识大纲和教程。二级基础知识的内容在原有的基础上做了重大调整，调整后的二级基础知识主要包括基本数据结构与算法、程序设计基础、软件工程基础、数据库设计基础四个部分。二级基础知识在各科笔试中占 30% 的题量。

本书共分 5 章，每一章的内容都是按照新大纲要求进行编写的。

为了使读者在最短的时间里，掌握考试要求的最基本的内容。第 1 章至第 4 章每章以基本知识点、出题要点、出题趋势、典型考题与解题范例、实战备考试题的形式编写。在应试要点中对大纲要求的知识点进行了全面的介绍。在典型考题与解题范例中以试题、试题考点、出题思路、题解答案、解题方法进行介绍，使考生对基本知识点深入了解。通过实战备考试题再次进行检查。在第 5 章中设计了 10 套笔试模拟试卷的实战演练，并对答案进行了分析。

第 1 章是基本数据结构与算法。主要内容是算法的基本概念、数据结构的基本概念及其定义；线性表及其基本运算、栈和队列及其基本运算、线性链表及其基本运算、二叉树的基本概念和存储结构及其遍历。最后还介绍了几种常用的查找与排序算法。

第 2 章是程序设计基础。主要内容介绍程序设计方法与风格、结构化程序设计、面向对象的程序设计方法。

第 3 章是软件工程基础。主要介绍软件工程基本概念、结构化分析方法、结构化设计方法、软件测试与调试。

第 4 章是数据库设计基础。主要介绍数据库、数据库管理系统、数据库系统的基本概念；数据模型、实体联系模型及 E-R 图等基本概念；关系代数理论中的基本运算；数据库设计的基本方法和步骤。

第 5 章是笔试模拟试卷与答案分析。设计了 10 套笔试模拟试卷。每套试卷的形式与实际考试时的笔试试卷相同：选择题 10 题，填空题 5 题。在每一套题后都对答案进行了分析。

在学习本教材时，建议初学者可以从头开始学习（应试指导中仅对统一公共基础知识要点进行介绍）或者系统学习统一的公共基础知识教程；对于已经具有有一些统一公共基础知识的读者，可以根据自己的情况选择，从典型考题与解题范例中，学习掌握应试技巧；从笔试模拟试卷中检查学习效果。

本书由蔡学望、田昆、刘金编著。在本书的编写过程中，蔡薇等参与了校对并提出了宝贵的修改意见。

由于编者水平有限，书中难免有错误或不妥之处，恳请读者批评指正。

编者
2005 年 1 月

目 录

第1章 基本数据结构与算法应试指导	1
1.1 基本数据结构与算法应试指导	2
1.1.1 算法的基本概念	2
1.1.2 算法的复杂度	4
1.2 数据结构的基本概念	4
1.2.1 数据结构的表示	6
1.2.2 数据结构类型	7
1.3 线性表及其顺序存储结构	7
1.3.1 线性表的基本概念	7
1.3.2 线性表的顺序存储及其运算	8
1.3.3 栈及其基本运算	9
1.3.4 队列及其基本运算	9
1.3.5 线性链表及其运算	11
1.4 树与二叉树	13
1.4.1 树的基本概念	13
1.4.2 二叉树及其遍历	13
1.5 查找	16
1.6 排序技术	16
1.6.1 交换类排序	16
1.6.2 插入类排序	17
1.6.3 选择类排序	18
1.7 典型考题与解题范例	19
1.8 本章实战备考试题	35
1.9 参考答案	38
第2章 程序设计基础应试指导	39
2.1 程序设计方法与风格应试指导	39
2.1.1 程序设计的风格	39
2.1.2 程序设计的方法	41
2.2 结构化程序设计应试指导	41
2.2.1 结构化程序的基本结构与特点	41
2.2.2 模块化程序设计	43
2.2.3 自顶向下、逐步细化的设计过程	43

2.3	面向对象的程序设计应试指导	44
2.3.1	面向对象技术的基本概念	44
2.3.2	面向对象技术的特点	45
2.4	典型考题与解题范例	46
2.5	本章实战备考试题	58
2.6	参考答案	59
第3章	软件工程基础应试指导	61
3.1	软件工程概述应试指导	62
3.1.1	软件工程的	62
3.1.2	软件生命周期	63
3.1.3	软件工具与软件开发环境	64
3.1.4	软件详细设计的表达	65
3.1.5	应用软件开发的原则和方法	67
3.2	结构化分析方法应试指导	68
3.2.1	需求分析与需求分析方法	68
3.2.2	结构化分析方法的特点	69
3.2.3	数据流图与数据字典	69
3.2.4	软件需求规格说明书	70
3.3	结构化设计方法	72
3.3.1	结构化设计方法的特点	72
3.3.2	结构化设计	74
3.3.3	由数据流图导出结构图	75
3.3.4	模块的耦合与内聚	75
3.4	软件测试与调试应试指导	76
3.4.1	软件测试的任务、原则和方法	76
3.4.2	软件测试技术	77
3.4.3	调试	78
3.4.4	软件测试的组成	79
3.4.5	软件测试的实施	80
3.4.6	测试报告	80
3.4.7	软件的维护	81
3.5	典型考题与解题范例	82
3.6	本章实战备考试题	102
3.7	参考答案	105
第4章	数据库设计基础应试指导	106
4.1	数据库的基本概念应试指导	107
4.1.1	数据管理技术的发展	107
4.1.2	数据库、数据库管理系统、数据库系统	108
4.1.3	数据库系统的内部结构体系	110

4.2	数据模型	111
4.2.1	数据模型的概念	111
4.2.2	数据模型	112
4.3	关系代数	114
4.4	数据库设计方法	116
4.4.1	数据库设计的基本概念	116
4.4.2	数据库设计的需求分析	117
4.4.3	数据库概念设计	118
4.5	典型考题与解题范例	121
4.6	本章实战备考试题	139
4.7	参考答案	142
第5章	笔试模拟试卷	144
5.1	第1套笔试模拟试卷	144
5.1.1	试题	144
5.1.2	答案与分析	146
5.2	第2套笔试模拟试卷	148
5.2.1	试题	148
5.2.2	答案与分析	150
5.3	第3套笔试模拟试卷	154
5.3.1	试题	154
5.3.2	答案与分析	155
5.4	第4套笔试模拟试卷	159
5.4.1	试题	159
5.4.2	答案与分析	160
5.5	第5套笔试模拟试卷	163
5.5.1	试题	163
5.5.2	答案与分析	165
5.6	第6套笔试模拟试卷	168
5.6.1	试题	168
5.6.2	答案与分析	169
5.7	第7套笔试模拟试卷	172
5.7.1	试题	172
5.7.2	答案与分析	174
5.8	第8套笔试模拟试卷	177
5.8.1	试题	177
5.8.2	答案与分析	178
5.9	第9套笔试模拟试卷	181
5.9.1	试题	181
5.9.2	答案与分析	183

5.10 第 10 套笔试模拟试卷.....	185
5.10.1 试题.....	185
5.10.2 答案与分析.....	187
参考书目	191

第 1 章

基本数据结构与算法应试指导

本章知识点

1. 算法的基本概念。
2. 数据结构的基本概念。
3. 线性表及顺序存储结构。
4. 栈和队列的顺序存储结构及其基本运算。
5. 链表的结构及其基本运算。
6. 树和二叉树。
7. 查找与排序。

本章出题点

1. 数据结构与算法的基本概念。
2. 线性表的顺序存储结构。
3. 栈和队列的顺序存储结构。
4. 线性链表的结构及其基本运算。
5. 二叉树的性质及遍历。
6. 顺序查找与二分法查找算法。
7. 基本排序算法。

本章出题趋势

1. 本章的考试分值为 12 分左右。
2. 一般以选择题 8 分，填空题 4 分出现。
3. 考试内容
 - (1) 算法与算法复杂度（时间复杂度、空间复杂度）。
 - (2) 逻辑结构和存储结构。
 - (3) 线性表的顺序存储及其操作。
 - (4) 链表的存储结构及其基本操作。
 - (5) 栈、队列的存储结构及其基本操作。

- (6) 非线性表（树、二叉树）的存储结构。
- (7) 二叉树的性质及遍历。
- (8) 顺序查找与二分法查找算法。
- (9) 交换排序（冒泡排序、快速排序）。
- (10) 选择排序（简单排序）。

1.1 基本数据结构与算法应试指导

1.1.1 算法的基本概念

一、算法

算法是应用项目研制软件的基础，也可以是计算机系统中为解决某个具体问题而建立的一套方法，即对解题方案的准确而完整的描述。算法分为数值型算法与非数值型算法两种。

算法不等于程序，也不等于计算方法。程序可以作为算法的一种描述，但程序通常还需考虑很多与方法和分析无关的细节问题，这是因为在编写程序时要受到计算机系统运行环境的限制。程序的编制不可能优于算法的设计。

基本的算法（操作）有插入、删除、替换、查找和排序。

二、算法的基本特征

定义在数据上的一组操作（算法）与数据结构是密切相关的，数据结构的运算（操作）是通过算法描述的。算法是对特定问题求解步骤的一种描述，它是指令的有限序列。一个算法必须具有以下 5 个特征。

- ① 可行性。算法中描述的操作都可以通过已经实现的基本运算执行有限次来实现。
- ② 确定性。一个算法中每一条指令必须有确切的含义，无二义性。
- ③ 有穷性。一个算法必须总是在执行有穷步之后结束，每一步都可在有穷时间内完成。
- ④ 输入。一个算法有零个或多个输入，这些输入取自于某个选定的对象的集合。
- ⑤ 输出。一个算法有一个或多个输出，这些输出是同输入有某些特定关系的量。

算法的含义与程序十分相似，又有区别。一个程序不一定满足有穷性，操作系统是一个程序，但正常时永不停止。算法中的指令没有机器必须执行的限制，算法若用机器可执行的语言来描述，则是一个程序。

三、算法的基本要素

算法的基本要素包括两点，一是对数据对象的运算和操作，二是算法的控制结构。

(1) 算法中对数据的运算和操作

对数据的运算和操作有算术运算、逻辑运算、关系运算和数据传输。

算法的描述可以采用语言、图形和表格方式。不管采用哪种方式，都是按解题要求从上

述基本操作中选择合适的操作组成解题的操作序列。

每一个算法是计算机可以执行的以指令的形式描述的基本操作。一个计算机系统能执行的所有指令的集合，称为指令系统。计算机程序就是按解题要求从计算机指令系统中选择合适的指令所组成的指令序列。

(2) 算法的控制结构

算法的控制结构是指算法中各操作之间的执行顺序。一个算法一般都可以由顺序、选择和循环三种基本控制结构组合而成。算法的基本框架，反映了算法设计结构化原则。

描述算法的工具通常有传统的流程图、N-S 结构化流程图和算法描述语言等。

四、算法设计基本方法

判断算法性能的标准是：①正确性；②可使用性；③可读性；④健壮性；⑤效率。

算法的效率主要指算法执行时计算机资源的消耗，包括存储和运行时间的开销，前者叫做算法的空间代价，后者叫做算法的时间代价（用时间复杂度衡量）。

算法设计基本方法包括以下几点。

(1) 列举法：根据提出的问题，列举所有可能的情况，并用问题中给定的条件检验哪些是需要的，哪些是不需要的。列举法的算法比较简单，常用于寻找路径、查找、搜索等实际问题。但当列举的可能情况较多时，必须优化方案，以减少运算工作量。

(2) 归纳法：通过列举少量的特殊情况，经过分析，最后找出一般的关系。归纳法要比列举法更能反映问题的本质，并且可以解决列举量为无限的问题。但是，从一个实际问题中总结归纳出一般的关系，并不是一件容易的事情，尤其是要归纳出一个数学模型更为困难。

(3) 递推法：递推法是指从已知的初始条件出发，逐次推出所要求的各中间结果和最后结果。递推法本质上属于归纳法，工程上许多递推关系式实际上是通过分析实际问题与归纳而得到的，在数值计算中是极为常见的。

(4) 递归：递归过程能将一个复杂的问题归结为若干个较简单的问题，然后将这些较简单的问题再归结为更简单的问题，这个过程可以一直做下去，直到最简单的问题为止。

递归的基础也是归纳。在工程实际中，有许多问题就是用递归来定义的，数学中的许多函数也是用递归来定义的。递归在可计算性理论和算法设计中占有很重要的地位。

递推与递归的实现方法不同，递推是从初始条件出发，逐次推出所需求的结果；而递归则是从算法本身到达递归边界的。通常，递归算法要比递推算法清晰易读，其结构比较简练。特别是在许多比较复杂的问题中，很难找到从初始条件推出所需结果的全过程，此时，设计递归算法要比递推算法容易得多。但递归算法的执行效率比较低。

递归分为直接递归与间接递归两种。如果一个算法 P 显式地调用自己则称为直接递归。如果算法 P 调用另一个算法 Q，而算法 Q 又调用算法 P，则称为间接递归调用。

(5) 减半递推技术：它是指工程上常用的分治法。所谓“减半”，是指将问题的规模减半，而问题的性质不变；所谓“递推”，是指重复“减半”的过程。

(6) 回溯法：在工程上，有些实际问题很难归纳出一组简单的递推公式或直观的求解步骤，并且也不能进行无限的列举。对于这类问题，一种有效的方法是“试”，通过对问题的分析，找出一个解决问题的线索，然后沿着这个线索逐步试探。对于每一步的试探，若试探成

功,就得到问题的解,若试探失败,就逐步回退,换别的路线再进行试探,这种方法称为回溯法。回溯法在处理复杂数据结构方面有着广泛的应用。

1.1.2 算法的复杂度

一、算法分析

算法分析的目的是确定一个算法的优劣。判断一个算法的优劣有五条准则。

算法的分析主要是分析算法所占用的计算机资源,即算法在计算机上运行时所占用的时间和存储空间的度量。

算法分析的结果一般是一个复杂的表达式,人们所关心的是它的上界,只给出它的数量级。算法的复杂度主要包括时间复杂度和空间复杂度两种。

二、算法的时间复杂度

算法的时间复杂度是指执行算法所需要的计算工作量。

算法的工作量用算法所执行的基本运算次数来度量,而算法所执行的基本运算次数是问题规模的函数,即

$$\text{算法的工作量} = f(n)$$

其中, n 是问题的规模。算法的时间度量记作

$$T(n) = O(f(n))$$

它表示随问题规模 n 的增大,算法执行时间增长率和 $f(n)$ 增长率相同, $O(n)$ 称为时间复杂度。算法中可能呈现的时间复杂度有:常量阶 $O(1)$ 、线性阶 $O(n)$ 、对数阶 $O(\log_2 n)$ 、平方阶 $O(n^2)$ 、指数阶 $O(2^n)$ 以及 $O(n^3)$ 、 $O(2n)$ 、 $O(n!)$ 等形式。 $O(1)$ 表示算法的运行时间为常量,它不随数据量 n 的改变而改变,如访问表中第一个元素时,无论该表的大小如何,其时间复杂度均为 $O(1)$ 。具有 $O(n)$ 数量级的算法被称为线性算法,其运行时间与 n 成正比,如对一个表进行顺序查找时,其时间复杂度就是 $O(n)$ 。各种不同数量级对应的值存在着如下关系:

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

三、算法的空间复杂度

对于算法所需存储空间的量度称为空间复杂度,记作

$$S(n) = O(f(n))$$

它是指执行这个算法所需要的内存空间,包括算法程序、输入的初始数据以及算法执行过程中所需要的附加空间。附加空间是指算法程序执行过程中的工作单元以及某种数据结构所需要的空间。在许多实际问题中,通常采用压缩存储技术,以便尽量减少不必要的附加空间。

1.2 数据结构的基本概念

数据是描述客观事物的数、字符以及所有能够被计算机识别、存储和加工处理的符号集合。

数据元素是数据的基本单位，即数据集合中的个体。在不同的条件下，数据元素又可称为元素、结点、顶点和记录等。一个数据元素可由若干个数据项组成，数据项是具有独立含义的数据的最小单位，有时也把数据项称为域或字段等。由记录所组成的线性表称为文件。

数据对象是具有相同特性的数据元素的集合，是数据的子集。被计算机加工的数据元素彼此之间存在某些联系，通常将数据元素间的这种联系关系称为结构。

数据结构就是带有结构特性的数据元素的集合。数据结构概念一般包括三个方面的内容：数据之间的逻辑结构、数据在计算机中的存储方式（也称为存储结构或物理结构）以及在这些数据上定义的运算的集合。

在数据处理领域中，每一个需要处理的对象都可以抽象成数据元素。在一个具有相同特征的数据元素集合中，各个数据元素之间存在有某种关系（即联系），这种关系反映了该集合中的数据元素所固有的一种结构。通常，把数据元素之间这种固有的关系简单地用前后件关系（或直接前驱与直接后继关系）来描述。例如，在考虑一年4个季节的顺序关系时，则“春”是“夏”的前件（即直接前驱），而“夏”是“春”的后件（即直接后继）。同样，“夏”是“秋”的前件；“秋”是“夏”的后件；“秋”是“冬”的前件，“冬”是“秋”的后件。

在考虑家庭成员间的辈分关系时，则“父亲”是“儿子”和“女儿”的前件，而“儿子”与“女儿”都是“父亲”的后件。

数据结构可以看作是相互间存在特定关系的数据元素的集合。

1. 数据的逻辑结构

数据的逻辑结构只对数据元素之间的逻辑关系进行抽象描述，而不考虑数据在计算机中的存储方式。

根据数据元素之间的关系的不同特性，数据的逻辑结构可分为线性结构和非线性结构。

（1）线性结构：数据元素之间的关系是一对一的，可用一个线性序列表示出来。线性表、数组、堆栈、队列和串等属于线性结构。

（2）非线性结构：树、二叉树和图等都属于非线性结构。

另外，还有数据元素之间是多对多关系的图形结构；数据元素间的关系是“属于同一个集合”的集合关系。

数据元素之间的逻辑关系，即为前后件关系。

2. 数据的存储结构

数据的存储结构是数据逻辑结构在计算机存储器里的实现。数据的存储结构也称数据的物理结构。数据结构在计算机中的映像（即表示）包括数据元素的映像和关系的映像两部分，而把表征某一数据结构特点及关系方式的基本信息单位称为该数据结构的结点。结点往往又分成几个域，每个域用来存放与这个结点有关的数据项的信息。

由于数据元素在计算机存储空间中的位置关系可能与逻辑关系不同，因此，为了表示存放在计算机存储空间中的各数据元素之间的逻辑关系（即前后件关系），在数据的存储结构中，不仅要存放各数据元素的信息，还需要存放各数据元素之间的前后件关系的信息。

根据元素间关系的不同表示方法，最主要的存储结构有以下几种。

顺序存储结构：数据元素之间的关系由存储单元的邻接关系来体现。

链式存储结构：每个数据元素中至少包含两个域，一个存放数据值的数据域，另一个是指针域，用指针来体现数据元素之间的逻辑关系。

除了通常采用的顺序存储和链式存储外,有时为了查找的方便还采用索引存储和散列存储。用不同的存储结构,其数据处理的效率是不同的。

如一年4个季节的数据结构中,“春”是“夏”的前件,“夏”是“春”的后件,但在对它们进行处理时,在计算机存储空间中,“春”这个数据元素的信息不一定被存储在“夏”这个数据元素信息的前面,有可能在后面,也可能不是紧邻在前面,而是中间被其他的信息所隔开。

因此,在进行数据处理时,选择合适的存储结构是很重要的。

3. 运算

数据的运算定义在数据的逻辑结构上,而实现是在存储结构上。主要的运算包括插入、删除、排序和查找等。

1.2.1 数据结构的表示

常用的各种数据结构有线性表、堆栈、队列和链表等。实现这些数据结构的各种算法,几乎都立足于计算机系统结构只提供按地址访问的一维线性存储器以及最基本、最简单的数据表示,数据表示指的是能由硬件直接辨认的数据类型。这样,这些数据结构要经过软件映像,变换成按地址访问一维存储器内的各种数据表示。如何用最少的存储空间存储这些数据结构,以及用什么样的算法能最快、最简地存取则是数据结构的研究课题。

1. 数据结构二元组表示法

数据结构B,为了反映D中各数据元素之间的前后件关系,一般用二元组来表示,即

$$B=(D,R)$$

例如,一年四季的数据结构可表示为:

$D=\{\text{春, 夏, 秋, 冬}\}$

$R=\{(\text{春, 夏}), (\text{夏, 秋}), (\text{秋, 冬})\}$

2. 图形表示法

在数据结构的图形表示中,对于数据集合D中的每一个数据元素用中间标有元素值的方框表示,一般称为数据结点,简称为结点。为了进一步表示各数据元素之间的前后件关系,对于关系R中的每一个二元组,用一条有向线段从前件结点指向后件结点。

例如,一年四季的数据结构可用如图1-1所示的图形来表示。

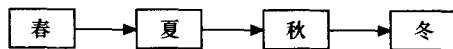


图 1-1 一年四季的图形表示

在数据结构中,没有前件的结点称为根结点;没有后件的结点称为终端结点(也称为叶子结点)。如一年四季的数据结构中,元素“春”所在的结点为根结点,“冬”为终端结点;在家庭成员间辈分关系的数据结构中,结点“父亲”为根结点,结点“儿子”与“女儿”均为终端结点如图1-2所示。

数据结构中除了根结点与终端结点外的其他结点一般称为内部结点。

结点是动态的,根据需要可以增加一个新结点(称为插入运算),

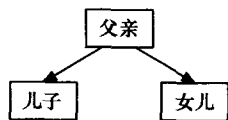


图 1-2 家庭成员的图形表示

也可以删除某个结点(称为删除运算)。插入与删除是对数据结构的两种基本运算,除此之外,对数据结构的运算还有查找、分类、合并、分解、复制和修改等。在对数据结构的处理过程中,不仅数据结构中的结点(即数据元素)个数在动态地变化,而且,各数据元素之间的关系也有可能在动态地变化。例如,一个无序表可以通过排序处理而变成有序表;一个数据结构中的根结点被删除后,它的某一个后件可能就变成了根结点;在一个数据结构中的终端结点后插入一个新结点,则原来的那个终端结点就不再是终端结点而成为内部结点了。

1.2.2 数据结构类型

计算机的基本数据类型有逻辑数、定点数、浮点数、十进制数、字符串和数组等。根据数据结构中各数据元素之间前后件关系的复杂程度,一般将数据结构分为:线性结构和非线性结构两大类。

在线性结构中,开始结点和终端结点都是惟一的,除了开始结点和终端结点外,其余结点都有且仅有一个前件,有且仅有一个后件,数据元素之间为一对一的关系,如顺序表就是典型的线性结构。一年四季的数据结构以及 n 维向量数据结构,它们都属于线性结构。

非线性结构又可以细分为树形结构和图形结构两类。

在树形结构中,每个结点最多只有一个前件,但可以有多多个后件,数据元素之间为一对多的关系,可以有多个终端结点。非线性结构的树形结构简称为树。

在图形结构中,每个结点的前件和后件的个数都可以是任意的,数据元素之间为多对多的关系。因此,可能没有开始结点和终端结点,也可能有多个开始结点、多个终端结点。图形结构简称为图。

线性结构是树形结构的特殊情况,而树形结构又是图形结构的特殊情况。

没有一个数据元素的数据结构称为空的数据结构。线性结构与非线性结构都可以是空数据结构。

1.3 线性表及其顺序存储结构

1.3.1 线性表的基本概念

线性表是一类数据结构的统称,栈结构、队列结构和链表结构等都是线性表。同一线性表中的元素必定具有相同特性,属于同一数据对象。线性表中所含元素的个数叫做线性表的长度,用 n 表示, $n \geq 0$ 。当 $n=0$ 时,表示线性表是一个空表,即表中不包含任何元素。线性表的逻辑结构如图 1-3 所示。

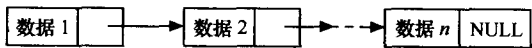


图 1-3 线性表的逻辑结构图

设序列中第 i 个元素为 a_i ($1 \leq i \leq n$)，则线性表的一般表示为：

$$(a_1, a_2, \dots, a_i, a_{i+1}, \dots, a_n)$$

其中， a_i 是数据对象的元素，通常称为线性表中的一个结点。数据元素在线性表中的位置只取决于它们自己的序号，即数据元素之间的相对位置是线性的。

对于一个非空线性表有如下一些结构特征：

- ① 有且只有一个根结点 a_1 ，它无前件；
- ② 有且只有一个终端结点 a_n ，它无后件；
- ③ 除根结点与终端结点外，其他所有结点有且只有一个前件，也有且只有一个后件。

线性表是一个具有相同特性的 n 个数据元素的有限序列。例如，26 个大写英文字母表，就是一个线性表。线性表是最简单、最基本、最常用的一类线性数据结构，使用相当灵活。它的基本操作是插入、删除、检索和排序等。

1.3.2 线性表的顺序存储及其运算

1. 线性表的顺序存储

用一段地址连续的存储单元，依次存储线性表里各元素的存储结构，称作线性表的顺序存储结构。它的特点是线性表的元素的逻辑顺序与其各元素的存储顺序是一致的。

这样，线性表中第 1 个元素的存移位置就是指定的存储位置，第 i 个元素 ($2 \leq i \leq n$) 的存储位置紧接在第 $i-1$ 个元素的存储位置后面。

假设线性表中的第 1 个数据元素的存储地址（指第 1 个字节的地址，即首地址）为 $ADR(a_1)$ ，每一个数据元素占 k 个字节，则线性表中第 i 个元素 a_i 在计算机存储空间中的存储地址为：

$$ADR(a_i) = ADR(a_1) + (i-1)k \quad (\text{线性表元素的地址访问公式})$$

即在顺序存储结构中，线性表中每一个数据元素在计算机存储空间中的存储地址由该元素在线性表中的位置序号惟一确定。

程序设计语言中的一维数组与计算机中实际的存储空间结构是类似的，这就便于用程序设计语言的一维数组对线性表进行各种运算处理。

在线性表的顺序存储结构下，可以对线性表进行各种处理。主要的运算有：插入、删除、查找、排序、分解、合并、复制和逆转等。

2. 顺序表的插入与删除运算

在一个顺序线性表中第 i ($1 \leq i \leq n$) 个元素之前插入一个新元素时，首先要从最后一个（即第 n 个）元素开始，直到第 i 个元素之间共 $n-i+1$ 个元素依次向后移动一个位置，腾出一个存储元素的空间，然后将新元素插入到第 i 项。插入结束后，线性表的长度就增加了 1。

要在一个顺序线性表中删除第 i ($1 \leq i \leq n$) 个元素时，则要从第 $i+1$ 个元素开始，直到第 n 个元素之间共 $n-i$ 个元素依次向前移动一个位置。删除结束后，线性表的长度就减少了 1。

线性顺序表的优点是便于数据的存取和查找，缺点是插入和删除运算需要移动大量的其他数据元素，效率比较低。