

21世纪高等院校计算机规划教材

盐城工学院
教材基金资助出版

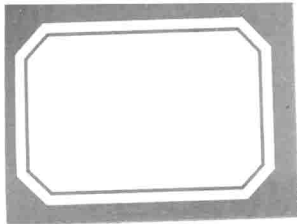
全国计算机等级考试教程

——二级公共基础与Visual Basic

邵洪成 董 琴 赵雪梅 朱锦新 编著



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



校计算机规划教材
盐城工学院教材基金资助出版

全国计算机等级考试教程

——二级公共基础与 Visual Basic

邵洪成 董 琴 赵雪梅 朱锦新 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是“二级公共基础”与“Visual Basic 程序设计”课程的配套教材,主要内容包括二级公共基础知识、Visual Basic 语言程序设计、全国计算机等级考试二级 VB 笔试模拟试题。书中有大量的题目,供给学生学习与训练。本书适合作为全国计算机等级考试二级 Visual Basic 的参考书。

图书在版编目(CIP)数据

全国计算机等级考试教程. 二级公共基础与 Visual Basic/
邵洪成等编著. — 北京: 中国铁道出版社, 2012. 7
21 世纪高等院校计算机规划教材
ISBN 978-7-113-14932-1

I. ①全… II. ①邵… III. ①电子计算机—水平考试
—教材②BASIC 语言—程序设计—水平考试—教材 IV.
①TP3

中国版本图书馆 CIP 数据核字(2012)第 144150 号

书 名: 全国计算机等级考试教程——二级公共基础与 Visual Basic
作 者: 邵洪成 董 琴 赵雪梅 朱锦新 编著

策 划: 张围伟	读者热线: 400-668-0820
责任编辑: 贾淑媛	特邀编辑: 彭丽群
编辑助理: 包 宁	
封面设计: 付 巍	
封面制作: 刘 颖	
责任印制: 李 佳	

出版发行: 中国铁道出版社(100054, 北京市西城区右安门西街 8 号)

网 址: <http://www.51eds.com>

印 刷: 航远印刷有限公司

版 次: 2012 年 7 月第 1 版 2012 年 7 月第 1 次印刷

开 本: 787mm×1092mm 1/16 印张: 19.5 字数: 477 千

印 数: 1~3 500 册

书 号: ISBN 978-7-113-14932-1

定 价: 38.00 元

版权所有 侵权必究

凡购买铁道版图书, 如有印制质量问题, 请与本社教材图书营销部联系调换。电话: (010) 63550836

打击盗版举报电话: (010) 63549504

前 言

随着计算机技术的飞速发展, 社会对大学生的计算机应用能力与软件开发水平的要求也在不断提高。“大学计算机基础”与“程序设计基础 (Visual Basic 程序设计)”是普通高等学校非计算机专业学生的必修课程。为进一步加强普通高等学校非计算机专业的“大学计算机基础”与“程序设计基础 (Visual Basic 程序设计)”课程的教学工作, 提高教学质量, 帮助学生顺利通过计算机等级考试, 提高学生的动手操作能力和计算机高级语言程序设计能力, 促进学生综合素质的提高, 根据全国计算机等级考试大纲, 并结合编者多年的教学经验, 编写了本书, 用于配合“大学计算机基础”与“程序设计基础 (Visual Basic 程序设计)”课程的教学, 同时也是在校大学生参加全国计算机等级考试不可多得的实用教材。读者可以到盐城工学院计算中心网站(<http://jszx.ycit.cn>)上下载相关素材, 也可以直接与编者联系。

本书内容分为 3 章: 第 1 章为二级公共基础知识; 第 2 章为 Visual Basic 语言程序设计; 第 3 章为全国计算机等级考试二级 VB 笔试模拟试题, 包括 8 套全真模拟试题。

本书由邵洪成、董琴、赵雪梅、朱锦新编著。本书的编者长期从事“大学计算机基础”与“程序设计基础 (Visual Basic 程序设计)”等课程的教学研究工作和指导学生进行计算机等级考试强化训练等教学实践工作, 有着丰富的教学经验, 并结合最新教学考试大纲, 紧扣教材, 根据教材和等级考试的重点、难点精心编写本书。全书的内容都与全国计算机等级考试二级 VB 相一致, 便于学生对所学的知识进行复习和巩固, 同时又能提高分析程序、编写程序的能力, 提高学生的动手操作能力, 最终能提高计算机等级考试的通过率。本书在编写过程中得到了盐城工学院教务处、实验教学部的领导及同行教师的关心和支持, 在此一并表示衷心的感谢!

由于编者水平有限, 书中难免有疏漏和不足之处, 敬请读者批评指正。

编者 E-mail: shc@ycit.cn

编 者

2012 年 5 月

目 录

第 1 章	二级公共基础知识	1
1.1	数据结构与算法	1
1.1.1	算法	1
1.1.2	数据结构	4
1.1.3	线性表及其顺序存储结构	5
1.1.4	栈和队列	6
1.1.5	线性链表	8
1.1.6	树与二叉树	9
1.1.7	查找技术	15
1.1.8	排序技术	15
	习题	16
1.2	程序设计基础	24
1.2.1	程序设计方法与风格	24
1.2.2	结构化程序设计	25
1.2.3	面向对象的程序设计	25
	习题	28
1.3	软件工程基础	30
1.3.1	软件工程基本概念	30
1.3.2	结构化分析方法	32
1.3.3	结构化设计方法	35
1.3.4	软件测试	39
1.3.5	程序调试	41
	习题	42
1.4	数据库设计基础	47
1.4.1	数据库的基本概念	47
1.4.2	数据模型	51
1.4.3	关系代数	55
1.4.4	数据库设计与管理	57
	习题	59
第 2 章	Visual Basic 语言程序设计	66
2.1	Visual Basic 语言概述	66
2.1.1	Visual Basic 语言	66
2.1.2	对象、属性、方法与事件	67
2.1.3	窗体的属性、方法与事件	67
2.1.4	控件	69

2.1.5	工程的保存、打开和运行	69
	习题	69
2.2	常用标准控件	72
	习题	83
2.3	Visual Basic 语言基础	89
2.3.1	数据类型	89
2.3.2	公共函数	92
2.3.3	运算符与表达式	96
2.3.4	算法与三种基本结构	98
2.3.5	赋值语句	98
2.3.6	InputBox()函数和 MsgBox()函数	99
	习题	100
2.4	选择结构	111
	习题	113
2.5	循环结构	118
	习题	120
2.6	数组	138
	习题	141
2.7	过程	162
	习题	166
2.8	菜单、键盘与鼠标事件	183
2.8.1	菜单的基本知识	183
2.8.2	键盘事件	186
2.8.3	鼠标事件	187
	习题	188
2.9	通用对话框	194
	习题	196
2.10	文件	202
	习题	205
第 3 章	全国计算机等级考试二级 VB 笔试模拟试题	214
	二级 VB 笔试模拟试题一	214
	二级 VB 笔试模拟试题二	224
	二级 VB 笔试模拟试题三	235
	二级 VB 笔试模拟试题四	244
	二级 VB 笔试模拟试题五	254
	二级 VB 笔试模拟试题六	264
	二级 VB 笔试模拟试题七	272
	二级 VB 笔试模拟试题八	283
	习题参考答案	294
	参考文献	304

1.1 数据结构与算法

1.1.1 算法

1. 算法的定义

所谓算法是指解题方案的准确而完整的描述，是一组严谨地定义运算顺序的规则，并且每一个规则都是有效的、明确的，此顺序将在有限的次数后终止。

2. 算法的基本特征

作为一个算法，一般应具有以下几个基本特征：

(1) 可行性

针对实际问题设计的算法，人们总希望能够得到满意的结果。但一个算法又总是在某个特定的计算工具上执行的，因此，算法在执行过程中往往要受到计算工具的限制，使执行结果产生偏差。在设计一个算法时，必须考虑它的可行性，否则不会得到满意的结果。

(2) 确定性

算法中的每个步骤都必须有明确定义，不允许有模棱两可的解释，也不允许有多义性。

(3) 有穷性

算法的有穷性是指算法必须能在有限的时间内做完，即算法必须能在执行有限个步骤后终止。

(4) 输入/输出性

输入/输出性又称拥有足够的情报，即一个算法是否有效，还取决于为算法所提供的输入信息是否足够。

3. 算法的基本要素

一个算法通常由两个基本要素组成：对数据的运算和操作、算法的控制结构。

(1) 算法中对数据的运算和操作

算法中对数据的基本运算和操作有以下四类：

① 算术运算：主要包括加、减、乘、除等运算。

② 逻辑运算：主要包括“与”、“或”、“非”等运算。

③ 关系运算：主要包括“大于”、“小于”、“等于”、“不等于”、“大于等于”、“小于等于”等运算。

④ 数据传输：主要包括赋值、输入、输出等操作。

(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。它包含顺序结构、选择（分支）结构和循环（重复）结构三种，如图 1-1-1 所示。

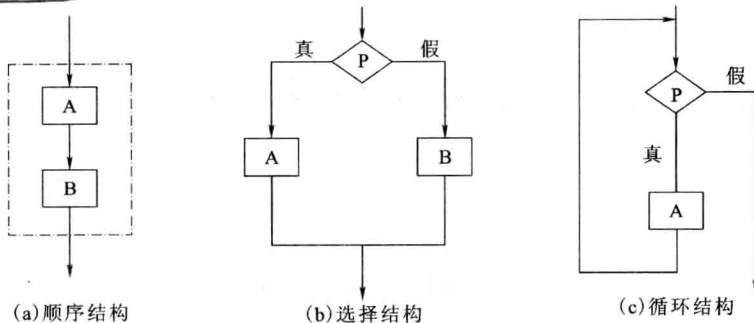


图 1-1-1 算法的三种基本结构

4. 算法设计基本方法

算法设计基本方法有列举法、归纳法、递推法、递归法和回溯法。

(1) 列举法

列举法的基本思想是根据提出的问题，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。

(2) 归纳法

归纳法的基本思想是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。归纳是一种抽象，即从特殊现象中找出一般关系。

(3) 递推法

所谓递推是指从已知的初始条件出发，逐次推出所要求的中间结果和最后结果。其中初始条件或是问题本身已经给定，或是通过问题的分析与化简而确定。

(4) 递归法

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。递归分为直接递归与间接递归两种。

直 接 递 归	间 接 递 归
<pre> Private Sub Test(x As Integer) ... Call Test(x) ... End Sub </pre>	<pre> Private Sub Test1(x As Integer) ... Call Test2(x) ... End Sub Private Sub Test2(x As Integer) ... Call Test1(x) ... End Sub </pre>

(5) 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。

5. 算法的复杂度

算法复杂度主要包含时间复杂度与空间复杂度。

(1) 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,即算法的时间复杂度是指算法执行过程中所需要的基本运算次数。

- ① 平均性态分析:是指用各种特定输入下的基本运算次数的加权平均值来度量算法的工作量。
- ② 最坏情况分析:是指在规模一定时,算法所执行的基本运算的最大次数。
- ③ 例题分析:

【例 1-1-1】

$x=x+1$

基本运算: x 增加 1。

基本运算的执行次数: 1。

时间复杂度: $O(1)$ 。

【例 1-1-2】

For $i=1$ To n

$x=x+1$

Next i

基本运算: x 增加 1。

基本运算的执行次数: n 。

时间复杂度: $O(n)$ 。

【例 1-1-3】

For $i=2$ To n

For $j=2$ To $i-1$

$x=x+1$

Next j

Next i

基本运算: x 增加 1。

基本运算的执行次数:

$i=2$ 0 次

$i=3$ 1 次

$i=4$ 2 次

$\vdots$ $\vdots$

$i=n$ $n-2$ 次

总次数 $= 1+2+3+\cdots+(n-2) = \frac{(n-1)(n-2)}{2} = \frac{n^2-n-2n+2}{2}$

时间复杂度: $O(\frac{n^2-n-2n+2}{2})$, 即 $O(n^2)$ 。

(2) 算法的空间复杂度

一个算法的空间复杂度一般是指执行这个算法所需要的内存空间。一个算法所占用的存储空间包括算法程序所占用的空间、输入的初始数据所占用的存储空间以及算法执行过程中所需要的额外空间。

1.1.2 数据结构

1. 数据结构研究和讨论的问题

数据结构主要研究和讨论以下三个方面的问题：

- ① 数据的逻辑结构。数据集合中各数据元素之间固有的逻辑关系。
- ② 数据的存储结构。对数据进行处理时，各数据元素在计算机中的存储关系。
- ③ 对各种数据结构进行的运算。

2. 数据元素及其联系

数据元素具有广泛的含义，一般来说，现实世界中客观存在的一切个体都可以是数据元素。在具有相同特征的数据元素集合中，各个数据元素之间存在着某种关系，这种关系反映了该集合中的数据元素所固有的一种结构。在数据处理领域中，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来描述。前后件关系是数据元素之间的一个基本关系，但前后件关系所表示的实际意义随具体对象的不同而不同。一般来说，数据元素之间的任何关系都可以用前后件关系来描述。

3. 数据的逻辑结构

线性
非线性

数据的逻辑结构是指反映数据元素之间逻辑关系的数据结构。数据的逻辑结构有两个要素：数据元素的集合、各数据元素之间的前后件关系。

4. 数据的存储结构

顺序、链接、索引

数据的逻辑结构在计算机存储空间中的存放方式称为数据的存储结构，也称为数据的物理结构。

一般来说，一种数据的逻辑结构根据需要可以表示成多种存储结构，常用的存储结构有顺序、链接、索引等存储结构。而采用不同的存储结构，其数据处理的效率是不同的。

5. 数据结构的图形表示

一个数据结构除了用二元关系表示外，还可以直观地用图形表示。在数据结构的图形表示中，对于数据集合中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，简称结点。为了进一步表示各数据元素之间的前后件关系，用一条有向线段从前件结点指向后件结点。

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（又称叶子结点）。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。

例如：描述一年四季的季节名如图 1-1-2 所示。其中，春是夏的前件，夏是春的后件，春是根，冬是终端结点（叶子结点）。

春	夏	秋	冬
---	---	---	---

图 1-1-2 结点前后件关系图

6. 线性结构与非线性结构

根据数据结构中各数据元素之间前后件关系的复杂程度，分为线性结构与非线性结构，如果一个非空的数据结构满足下列两个条件：①有且仅有一个根结点；②每一个结点最多有一个前件，也最多有一个后件，则称该数据结构为线性结构，又称线性表。

如果一个数据结构不是线性结构,则称为非线性结构,如图1-1-3所示。对非线性结构的存储与处理比线性结构要复杂得多。

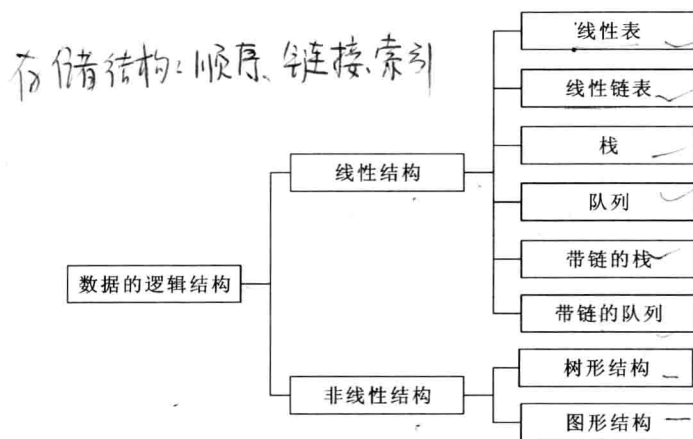


图 1-1-3 线性结构与非线性结构

1.1.3 线性表及其顺序存储结构

1. 线性表的概念

线性表由一组数据元素构成,数据元素之间的相对位置是线性的。非空线性表 $L=\{a_1, a_2, a_3, \dots, a_n\}$ 的结构特征有:

- 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。结点个数 n 称为线性表的长度, 当结点个数为 0 时, 称为空表。

2. 线性表顺序存储结构的基本特点: 按逻辑顺序存放的。

- ① 线性表中数据元素类型一致, 只有数据域, 存储空间利用率高。
- ② 所有元素所占的存储空间是连续的。
- ③ 各数据元素在存储空间中是按逻辑顺序依次存放的。
- ④ 做插入、删除操作时需移动大量元素。
- ⑤ 空间估计不明时, 按最大空间分配。

3. 线性表的主要运算

在线性表的顺序存储结构下, 可以对线性表进行各种处理。主要运算有:

- ① 线性表的插入: 在线性表的指定位置处添加一个新的元素。
- ② 线性表的删除: 在线性表中删除指定的元素。
- ③ 线性表的查找: 在线性表中查找某个(或某些)特定的元素。
- ④ 线性表的排序: 对线性表中的元素进行排序。
- ⑤ 线性表的分解: 按要求将一个线性表分解成多个线性表。
- ⑥ 线性表的合并: 按要求将多个线性表合并成一个线性表。

【例 1-1-4】线性表的插入运算, 如图 1-1-4 所示。

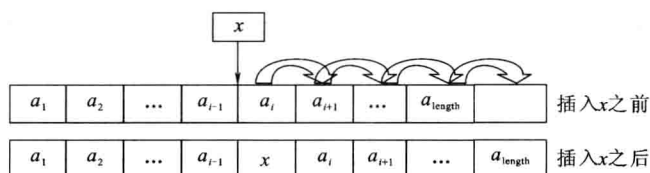


图 1-1-4 线性表的插入

【插入算法分析】假设线性表中含有 n 个数据元素，在进行插入操作时，若假定在 $n+1$ 个位置上插入元素的可能性均等，则平均移动元素的个数为：

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

【例 1-1-5】线性表的删除运算，如图 1-1-5 所示。

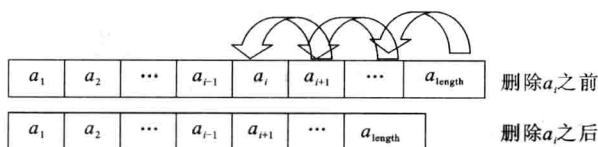


图 1-1-5 线性表的删除

【删除算法分析】在进行删除操作时，若假定删除每个元素的可能性均等，则平均移动元素的个数为：

$$\frac{1}{n} \sum_{i=1}^n (n-i) = \frac{n-1}{2}$$

由此可见，顺序存储结构表示的线性表，在做插入或删除操作时，平均需要移动一半的数据元素。

1.1.4 栈和队列

1. 栈及其基本运算

(1) 栈的基本概念

栈是一种特殊的线性表，其插入与删除操作都只在线性表的一端进行。在栈中，允许插入与删除的一端称为栈顶，而不允许插入与删除的一端称为栈底。栈顶元素总是最后被插入的元素，也是最先被删除的元素；栈底元素总是最先被插入的元素，也是最后才被删除的元素（假设栈 $s=(a_1, a_2, \dots, a_n)$ ，如图 1-1-6 所示）。栈是按照“先进后出（FILO）”或“后进先出（LIFO）”的原则组织数据的。

栈的描述：

- ① 栈：是限定仅在表尾进行插入或删除操作的线性表。
- ② 栈顶：表尾。
- ③ 栈底：表头。
- ④ 空栈：不含元素的空表。

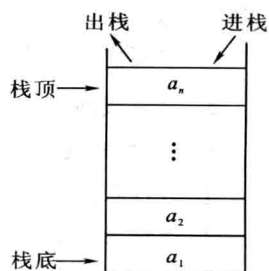


图 1-1-6 栈

(2) 栈的运算

通常用指针 top 表示栈顶的位置, 用指针 $bottom$ 指向栈底, 向栈中插入一个元素称为入栈运算, 从栈中删除一个元素称为退栈运算, 栈顶指针 top 动态反映了栈中元素的变化情况。在栈的顺序存储空间 $S(1:m)$ 中, $S(bottom)$ 称为栈底元素, $S(top)$ 称为栈顶元素, $top=0$ 表示栈空, $top=m$ 表示栈满, 如图 1-1-7 所示。

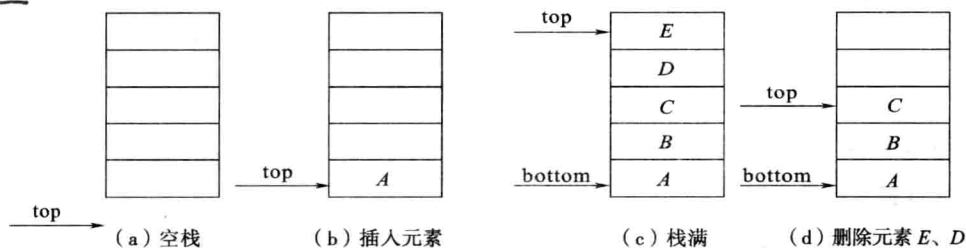


图 1-1-7 栈的动态示意图

2. 队列及其基本运算

(1) 队列的基本概念

队列是指允许在一端进行插入, 在另一端进行删除的线性表。允许插入的一端称为队尾, 通常用一个队尾指针 $rear$ 指向最后被插入的元素; 允许删除的一端称为队头 (排头), 通常用一个排头指针 $front$ 指向排头元素的前一个位置 (假设队列为 $a=(a_1, a_2, \dots, a_n)$, 如图 1-1-8 所示)。在队列这种数据结构中, 最先插入的元素最先被删除, 反之, 最后插入的元素将最后被删除。因此, 队列是“先进先出 (FIFO)”或“后进后出 (LILO)”的线性表, 它体现了“先来先服务”的原则。

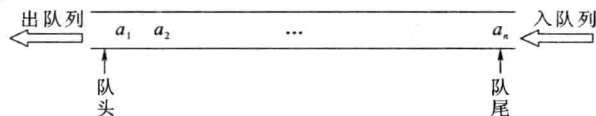


图 1-1-8 队列

(2) 循环队列的运算

① 入队运算。入队运算是指在循环队列的队尾加入一个新元素。这个运算有两个基本操作: 首先将队尾指针加 1; 然后将新元素插入到队尾指针指向的位置。当循环队列非空且队尾指针等于排头指针时, 说明循环队列已满, 不能进行入队运算, 若此时进行入队运算, 这种情况称为“上溢”。

② 退队运算。退队运算是指在循环队列的排头位置退出一个元素并赋值给指定的变量。这个运算有两个基本操作: 首先将排头指针加 1, 然后将排头指针指向的元素赋给指定的变量。当循环队列为空时, 不能进行退队运算, 若此时进行退队运算, 这种情况称为“下溢”。

③ 循环队列的基本思想。是指把队列设想为一个循环的表, 即 $elem[0]$ 接在 $elem[maxsize-1]$ 之后, 如图 1-1-9 所示。

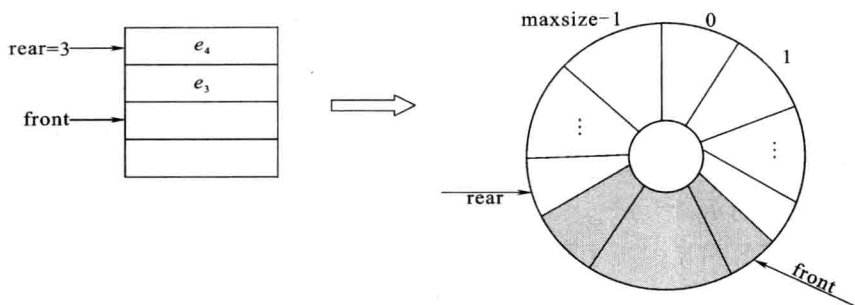


图 1-1-9 循环队列

1.1.5 线性链表

1. 线性链表的基本概念

线性表的链式存储结构称为线性链表。为了存储线性表中的每一个元素，一方面要存储数据元素的值，另一方面要存储各数据元素之间的前后件关系。为此，将存储空间中的每一个存储结点分为两部分：一部分用于存储数据元素的值，称为数据域，另一部分用于存放下一个数据元素的存储序号，即指向后件结点，称为指针域。在线性链表中，用一个专门的指针 head 指向线性链表中的第一个数据元素的结点，线性表中最后一个元素没有后件，因此，线性链表中最后一个结点的指针为空，用 NULL 或 0 表示，表示链表终止。

在线性链表中，各数据元素之间的前后件关系是由各结点的指针域来指示的，指向线性表中第一个结点的指针 head 称为头指针，当 head=NULL（或 0）时称为空表。对于线性链表，可以从头指针开始，沿各结点的指针扫描到链表中的所有结点。

① 线性单链表。在这种链表中，每一个结点只有一个指针域，由这个指针只能找到后件结点，但不能找到前件结点，即在线性单链表中，只能顺指针向链尾方面进行扫描，如图 1-1-10 所示。

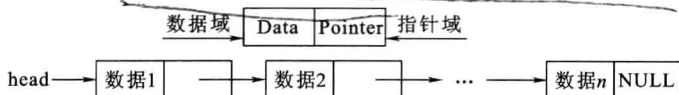


图 1-1-10 线性单链表

② 双向链表。线性链表中的每个结点设置两个指针，一个称为左指针（Link），用以指向其前件结点，另一个称为右指针（Rlink），用以指向其后件结点，如图 1-1-11 所示。

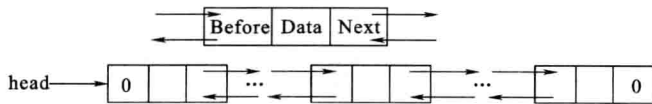


图 1-1-11 双向链表

2. 线性链表的基本运算

(1) 在线性链表中查找指定元素

在线性链表中查找包含指定元素值的前一个结点，当找到包含指定元素的前一个结点，就可以在该结点后插入新结点或删除该结点后的一个结点。

(2) 线性链表的插入

在线性链表中插入一个新元素，首先要给该元素分配一个新结点，以便用于存储该元素的值，新结点可以从可利用栈中取得，然后将存放新元素值的结点链接到线性链表中指定的位置。

【例 1-1-6】单链表的插入运算，如图 1-1-12 所示。

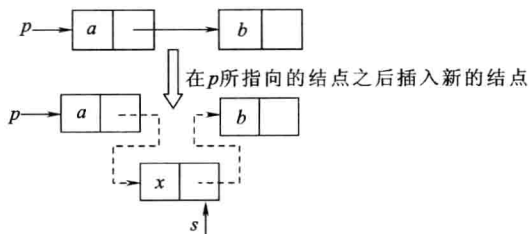


图 1-1-12 单链表的插入

(3) 线性链表的删除

在线性链表中删除包含指定元素的结点，首先要在线性链表中找到这个结点，然后将要删除的结点放回到可利用栈。

【例 1-1-7】单链表的删除运算，要求删除结点 a_i ，如图 1-1-13 所示。

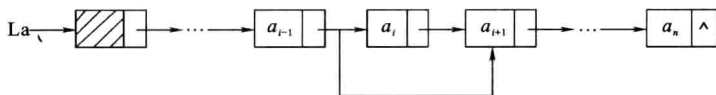


图 1-1-13 单链表的删除

3. 循环链表及其基本运算

循环链表（见图 1-1-14）的结构与线性链表相比，具有以下两个特点：

- ① 在循环链表中增加了一个表头结点，循环链表的头指针指向表头结点。
- ② 循环链表中最后一个结点的指针域不是空，而是指向表头结点。

循环链表的基本运算主要有插入和删除，循环链表的插入和删除的方法与线性单链表基本相同。

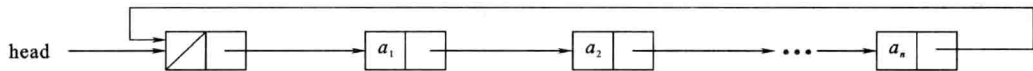


图 1-1-14 循环链表

1.1.6 树与二叉树

1. 树的基本概念

树是一种简单的非线性结构，所有数据元素之间的关系具有明显的层次特性，如图 1-1-15 所示。有关树结构的基本术语如下：

① 根结点与叶子结点：在树结构中，每个结点只有一个前件，称为父结点，没有前件的结点只有一个，称为树的根结点。在树结构中，每一个结点可以有多个后件，它们都称为该结点的子结点，没有后件的结点称为叶子结点。

② 结点的度：在树结构中，一个结点所拥有的后件个数称为该结点的度。

③ 树的度：在树中，所有结点中的最大的度称为树的度。

④ 树的深度：树的最大层次称为树的深度。在树中，以某结点的一个子结点为根构成的树称为该结点的一棵子树。在树中，叶子结点没有子树。

树结构具有明显的层次关系，即树是一种层次结构。在树结构中，一般按如下原则分层：根结点在第 1 层。同一层上所有结点的所有子结点都在下一层。

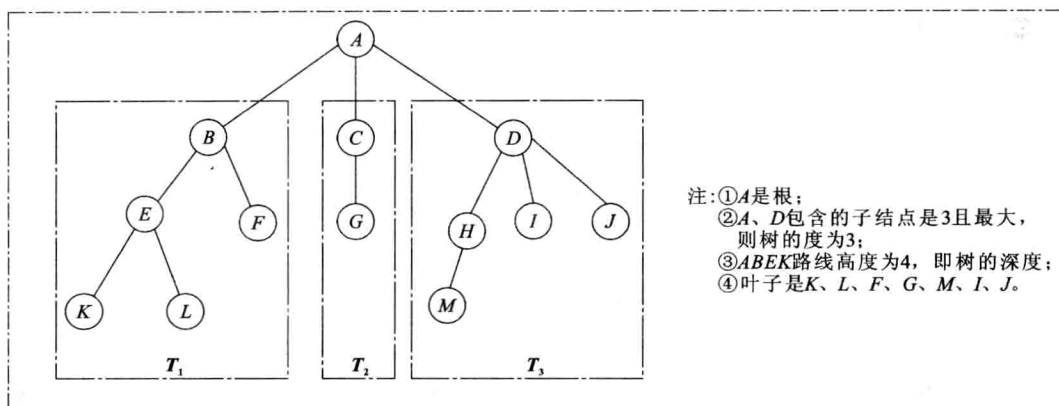


图 1-1-15 树

2. 二叉树及其基本性质

(1) 二叉树的特点

- ① 非空二叉树只有一个根结点。
- ② 每一个结点最多有两棵子树, 且分别称为该结点的左子树与右子树。

在二叉树中, 每一个结点的度最大为 2, 即所有子树(左子树或右子树)也均为二叉树, 而树结构中的每一个结点的度可以是任意的。另外, 二叉树中的每一个结点的子树被明显地分为左子树与右子树。在二叉树中, 一个结点可以只有左子树而没有右子树, 也可以只有右子树而没有左子树。当一个结点既没有左子树也没有右子树时, 该结点即是叶子结点。

(2) 二叉树的基本性质

【性质 1】在二叉树的第 k 层上, 最多有 2^{k-1} 个结点。

如图 1-1-16 所示的二叉树上:

第一层上 ($i=1$), 有 $2^{1-1}=1$ 个结点。

第二层上 ($i=2$), 有 $2^{2-1}=2$ 个结点。

第三层上 ($i=3$), 有 $2^{3-1}=4$ 个结点。

第四层上 ($i=4$), 有 $2^{4-1}=8$ 个结点。

【性质 2】深度为 k 的二叉树最多有 2^k-1 个结点。

$$2^0 + 2^1 + 2^2 + \cdots + 2^{k-1} = 2^k - 1$$

【性质 3】在任意一棵二叉树中, 度为 0 的结点 (即叶子结点) 总是比度为 2 的结点多一个, 如图 1-1-17 所示。

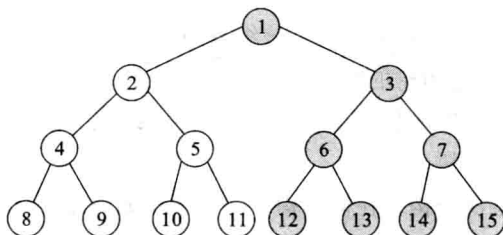


图 1-1-16 二叉树一

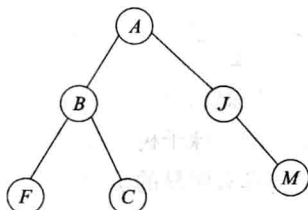


图 1-1-17 二叉树二

【性质 4】具有 n 个结点的二叉树，其深度 k 至少为 $\lfloor \log_2 n \rfloor + 1$ ，其中 $\lfloor \log_2 n \rfloor$ 表示取 $\log_2 n$ 的整数部分（见图 1-1-18）。

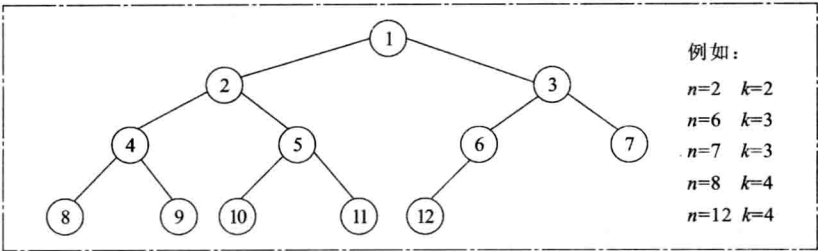


图 1-1-18 二叉树三

（3）满二叉树与完全二叉树

满二叉树与完全二叉树是两种特殊形态的二叉树。

① 满二叉树。所谓满二叉树是指，除最后一层外，每一层上的所有结点都有两个子结点，如图 1-1-19 所示。也就是说，在满二叉树中，每一层上的结点数都达到最大值，即在满二叉树的第 k 层上有 2^{k-1} 个结点，且深度为 m 的满二叉树有 $2^m - 1$ 个结点。

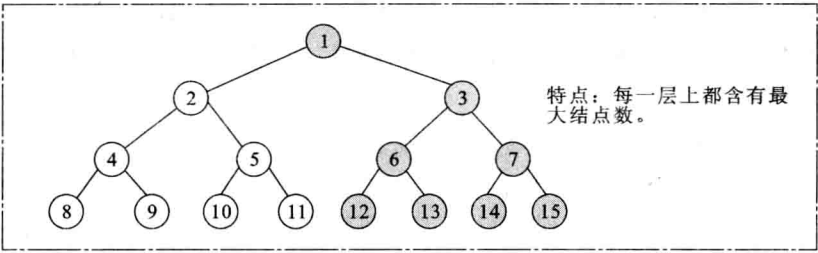


图 1-1-19 满二叉树

② 完全二叉树。所谓完全二叉树是指，除最后一层外，每一层上的结点数均达到最大值，在最后一层上只缺少右边的若干结点，如图 1-1-20 所示。

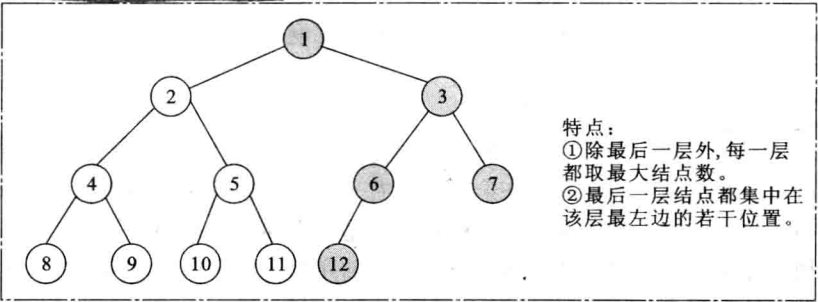


图 1-1-20 完全二叉树

更确切地说，如果从根结点起，对二叉树的结点自上而下、自左至右用自然数进行连续编号，则深度为 m 、且有 n 个结点的二叉树，当且仅当其每一个结点都与深度为 m 的满二叉树中编号从 $1 \sim n$ 的结点一一对应时，称为完全二叉树。

对于完全二叉树来说，叶子结点只可能在层次最大的两层上出现；对于任何一个结点，若其右分支下的子孙结点的最大层次为 p ，则其左分支下的子孙结点的最大层次为 p 或 $p+1$ 。

满二叉树也是完全二叉树，而完全二叉树一般不是满二叉树。