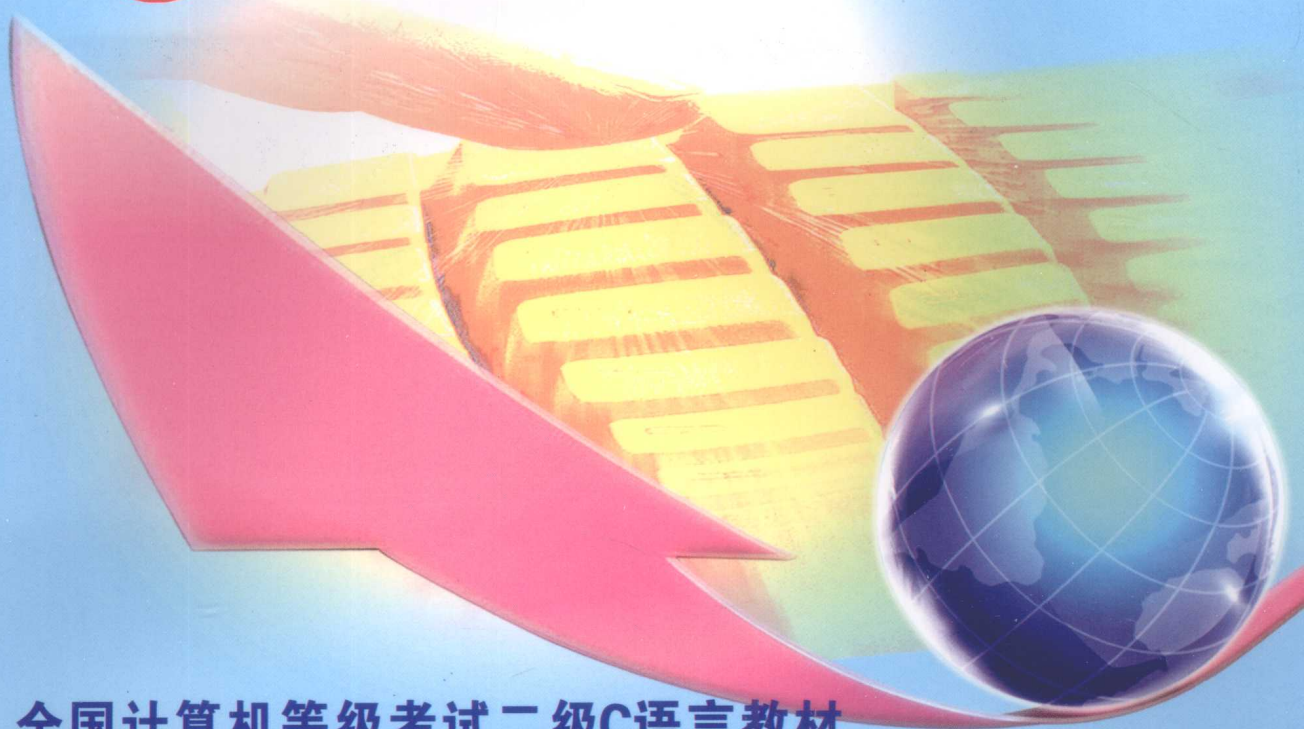




天行健教育

www.edu-exam.net



全国计算机等级考试二级C语言教材

公共基础

National Computer Rank Examination

天行健教研组 编

天行健教育专用教材



江西科学技术出版社

全国计算机等级考试二级 C 语言教材

公 共 基 础

主 编：蒋金良

编 委：（按姓氏笔画排名）

丁宁虹 付中华 刘 华 李小明

张 侃 邱 霞 姚玉生 徐忠良

温润良 赖华平

 江西科学技术出版社

图书在版编目 (CIP) 数据

全国计算机等级考试二级 C 语言教材. 公共基础 / 蒋金良编. —南昌:

江西科学技术出版社, 2010. 8

ISBN 978-7-5390-3990-9

I. ①全… II. ①蒋… III. ①C 语言—程序设计—水平考试—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2010) 第 167975 号

国际互联网 (Internet) 地址:

<http://www.jxkjcb.com>

选题序号: ZK2010533

图书代码: B10023-101

全国计算机等级考试二级 C 语言教材 公共基础

天行健教研组 编

出版发行 江西科学技术出版社

社 址 南昌市蓼洲街 2 号附 1 号

邮编: 330009 电话: (0791) 6623491 6639342 (传真)

印 刷 江西和平印刷厂

经 销 各地新华书店

开 本 188mm×265mm 1/16

字 数 46 千字

印 张 3.875

版 次 2010 年 8 月第 1 版 2010 年 8 月第 1 次印刷

书 号 ISBN 978-7-5390-3990-9

定 价 128.00 元 (共五册)

赣版权登字-03-2010-283

版权所有, 侵权必究

本书凡属印装质量问题, 可向承印厂调换

前言

全国计算机等级考试 (National Computer Rank Examination, NCRE), 是用于考查应试人员计算机应用知识与技能的全国性计算机水平考试体系。它是由教育部考试中心主办, 是计算机类考试中规模最大、考试人数最多的考试。全国每年有超过 1000 万考生参加这个考试, 它也是排在高考和大学英语等级考试之后的全国第三大考试。通过 NCRE 考试, 不仅能够掌握丰富的计算机知识, 证明自身的能力与价值, 同时也可以为今后工作、晋升和深造打下良好的基础。例如, 在职称评定中, 计算机等级考试的证书可以直接作为计算机能力水平的认证。

本书是针对全国计算机等级考试二级 C 语言科目所编写的辅导教材。计算机二级考试的笔试中, 程序设计部分占了 70%, 还有 30% 是公共基础知识部分, 这部分对于所有的二级科目而言都是相同的。公共基础知识的考点尽管难度不大, 但涉及面广, 包含了数据结构、算法、程序设计基础知识、软件工程和数据库技术。因此, 考生还是需要花一定的时间来认真学习。

为了帮助考生能够以最少的精力掌握这些公共基础知识, 我们组织中国科技大学、合肥工业大学、西安交通大学、贵州大学等重点高校的一线 C 语言教师, 在分析历年真题的基础上, 严格遵照考试大纲的要求, 编写了这本公共基础知识的教程, 力求以最简明的语言和篇幅囊括考试大纲要求的所有考点。同时, 我们也对知识点进行了总结, 将其中最重要的部分编制成习题, 以帮助考生更好地学习和记忆。

本书在编写过程中得到了许多高校教师和等级考试辅导专家的大力支持, 为本书的编写提出了许多宝贵的意见, 在此表示衷心的感谢。由于编者水平有限, 书中难免有缺漏不当之处, 热切期望得到广大专家和读者的批评指正!

天行健教研组
2010 年 8 月

目 录

第1章 数据结构与算法	1
1.1 算法.....	1
1.1.1 什么是算法.....	1
1.1.2 算法的复杂度.....	2
1.2 数据结构的基本概念.....	2
1.2.1 什么是数据结构.....	2
1.2.2 数据结构的图形表示.....	3
1.2.3 线性结构和非线性结构.....	4
1.3 线性表及其顺序存储结构.....	4
1.3.1 线性表的基本概念.....	4
1.3.2 线性表的顺序存储结构.....	5
1.3.3 线性表的插入和删除运算.....	5
1.4 栈和队列.....	6
1.4.1 栈及其基本运算.....	6
1.4.2 队列及其基本运算.....	6
1.5 线性链表.....	7
1.5.1 线性链表的基本概念.....	7
1.5.2 线性链表的基本运算.....	8
1.5.3 循环链表及其基本运算.....	8
1.6 树与二叉树.....	9
1.6.1 树的基本概念.....	9
1.6.2 二叉树及其基本性质.....	10
1.6.3 二叉树的存储结构.....	12
1.6.4 二叉树的遍历.....	13
1.7 查找技术.....	13
1.7.1 顺序查找.....	14
1.7.2 二分法查找.....	14
1.8 排序技术.....	14
第2章 程序设计基础	18
2.1 程序设计方法和风格.....	18
2.2 结构化程序设计.....	18
2.2.1 结构化程序设计方法的主要原则.....	18
2.2.2 结构化程序的基本结构.....	19

2.3	面向对象的程序设计	19
2.3.1	面向对象方法的必要性	19
2.3.2	面向对象方法的基本概念	19
2.3.3	面向对象方法的主要优点	19
第3章	软件工程基础	21
3.1	软件工程基本概念	21
3.1.1	软件工程的相关概念	21
3.1.2	软件危机与软件工程	21
3.1.3	软件生命周期	22
3.1.4	软件开发工具与软件开发环境	22
3.2	结构化分析方法	22
3.2.1	需求分析及分析方法	23
3.2.2	结构化分析方法	23
3.2.3	软件需求规格说明书(SRS)	23
3.3	结构化设计方法	24
3.3.1	软件设计的基础	24
3.3.2	概要设计和详细设计	24
3.4	软件测试	26
3.4.1	软件测试的定义和目的	26
3.4.2	软件测试方法	26
3.4.3	软件测试过程的步骤	27
3.5	程序的调试	28
第4章	数据库设计基础	30
4.1	数据库系统的基本概念	30
4.1.1	数据、数据库、数据库管理系统	30
4.1.2	数据库系统的发展及特点	30
4.1.3	数据库系统的内部结构体系	31
4.2	数据模型	31
4.2.1	数据模型的基本概念	31
4.2.2	实体联系模型及E-R图	32
4.2.3	常见数据模型	33
4.3	关系运算	34
4.3.1	关系的数据结构及运算	34
4.3.2	关系代数的扩充运算	35
4.4	数据库设计方法和步骤	36
第5章	公共基础知识习题	39
附录	公共基础知识总结	47

第1章 数据结构与算法

【主要内容】

- 算法的基本概念
- 基本数据结构：线性表、栈、队列、树、二叉树
- 查找算法 排序算法

【应试指导】

- 算法的基本概念、数据结构的定义、栈和树几乎是每次必考的知识点
- 查找和排序基本上每年一个题目
- 线性表、队列和线性链表经常与其他知识点结合出题。

1.1 算法

简单的说，算法是解决问题的方法和步骤。本小节从算法的定义和基本特征以及衡量一个算法优劣的时间和空间复杂度来论述。

1.1.1 什么是算法

1. 算法

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。

算法不等于程序，也不等于计算机方法，因而程序的编制不可能优于算法的设计。

2. 算法的基本特征

- (1) 可行性：算法中的操作能够用已经实现的基本运算执行有限次来实现。
- (2) 确定性：对需要执行的每一步操作，必须给出清楚，严格的规定。
- (3) 有穷性：算法必须在有限的时间内完成。
- (4) 拥有足够的情报：一个算法执行的结果总是与输入变量的初始数据有关，不同的输入

将会有不同的输出。当输入不够或输入错误时，算法将无法执行或报错。例如， $a=4$ ， $b=5$ ，求 $a*b*c$ 的值，由于没有对 c 进行初始化，无法计算出正确的答案，所以算法只有在拥有足够的输入信息和初始化信息时，才是可行的。

3. 算法的组成要素

对数据对象的运算和操作主要包括算术运算、逻辑运算、关系运算以及数据传输操作。

- (1) 算术运算：主要包括“+”、“-”、“ $\times$ ”、“ $\div$ ”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“ $>$ ”、“ $<$ ”、“ $=$ ”、“ $\neq$ ”等运算。
- (4) 数据传输：主要包括赋值、输入、输出等操作。

4. 算法设计的基本方法

(1) 列举法 (2) 归纳法 (3) 递推法 (4) 递归法 (5) 减半递推技术

1.1.2 算法的复杂度

算法复杂度包括时间复杂度和空间复杂度，用来衡量一个算法的优劣。

(1) 算法的时间复杂度是指执行算法所需要的计算工作量，可以用执行算法的过程中所需基本运算的执行次数来度量。

(2) 算法的空间复杂度指执行这个算法所需要的内存空间。

1.2 数据结构的基本概念

在我们进行应用程序开发时，需要处理的数据量很大，数据类型也很多，如何把这些数据有机地存储在计算机中，即怎样组织这些数据，这就是本节讨论的问题。

1.2.1 什么是数据结构

1. 数据结构

数据结构是计算机科学与技术领域广泛使用的一个基本术语，用来反映数据的内部组成。在给出数据结构的定义前，我们先给出几个概念和术语。

◆数据是对客观事物的符号表示，在计算机科学中，指所有能输入到计算机并被计算机处理的符号的总称，如数字、文本、声音、图像、视频等。

◆数据元素是数据的基本单位，一般由若干数据项构成。数据项是数据的最小单位。

数据结构的定义：数据结构是一个二元组 $\text{Data_Structure}(D, R)$ ，其中， D 是一个数据元素的有限集合， R 是 D 上的关系的有限集合，它反映了 D 中各数据元素之间的前后件关系，前后件关系可以用一个二元组来表示。

2. 数据结构的研究内容

(1) 数据的逻辑结构，即数据集合中各数据元素之间所固有的逻辑关系。数据的逻辑结构包含：

①表示数据元素的信息；

②表示各数据元素之间的前后件关系。

(2) 数据的存储结构，即在对数据进行处理时，各数据元素在计算机中的存储方式。数据的存储结构有顺序、链接、索引等多种存储结构。

①顺序存储。它是把逻辑上相邻的结点存储在物理位置相邻的存储单元里，结点间的逻辑关系由存储单元的邻接关系来体现。线性表的顺序存储结构具有以下两个基本特点：

▲线性表中所有元素的所占的存储空间是连续的；

▲线性表中各数据元素在存储空间中是按逻辑顺序依次存放的。 a_i 的存储地址为： $\text{adr}_{(a_i)} = \text{adr}_{(a_1)} + (i-1) * L - \text{adr}_{(a_1)}$ 为第一个元素的地址， L 代表每个元素所占的字节数。

存储地址	数据元素	线性表中位序
$LOC(a_1)$	a_1	1
$LOC(a_1)+L$	a_2	2
$LOC(a_1)+2*L$	a_3	3
$\vdots$	$\vdots$	$\vdots$
$LOC(a_1)+(i-1)*L$	a_i	i
$\vdots$	$\vdots$	$\vdots$
$LOC(a_1)+(n-1)*L$	a_n	n

图 1.1 数据的顺序存储示意图

②链式存储。它不要求逻辑上相邻的结点在物理位置上亦相邻，结点间的逻辑关系是由附加的指针字段表示的。由此得到的存储表示称为链式存储结构。

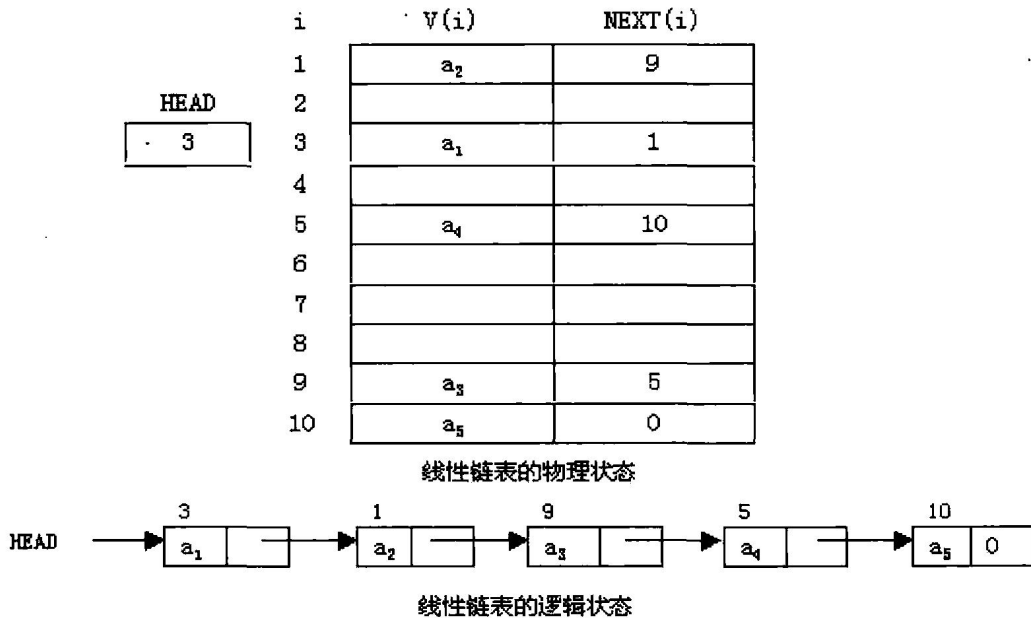


图 1.2 线性链表的表示

1.2.2 数据结构的图形表示

根据数据元素之间的关系的不同，通常可以分为 4 种结构：集合、线性结构、树型结构和图状结构。用图形直观地表示这些结构，如图 1.3 所示：

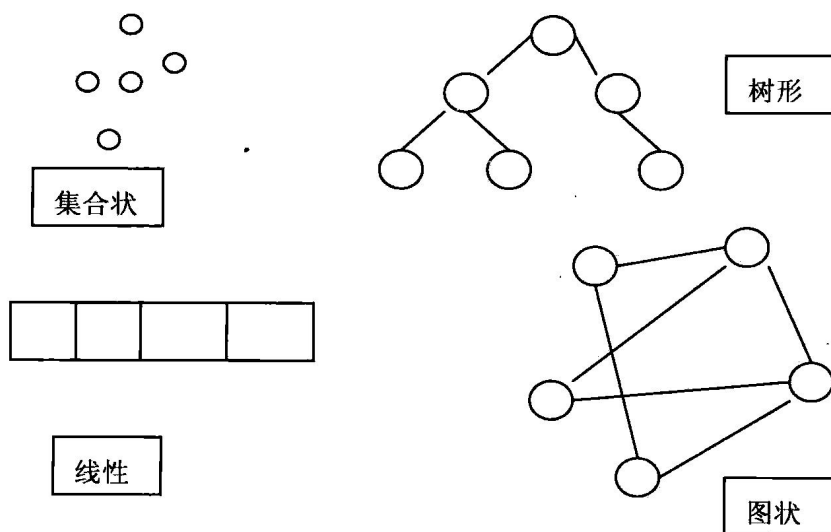


图 1.3 数据结构的图形表示

1.2.3 线性结构和非线性结构

(1) 线性结构

条件:

- 有且只有一个根结点（没有前件的结点）
- 每一个结点最多有一个前件，也最多有一个后件。

常见的线性结构有线性表、栈、队列和线性链表等。

(2) 非线性结构 不满足线性结构条件的数据结构。

常见的非线性结构有树、二叉树和图等。

1.3 线性表及其顺序存储结构

线性表是一种非常重要的数据结构，本小节从线性表的定义谈起，重点介绍线性表的插入和删除运算。

1.3.1 线性表的基本概念

线性表是一种最常见的线性数据结构。一个线性表是若干个数据元素构成的有限序列。在非空线性表中，存在唯一的一个开始元素和一个末尾元素，除了这两个元素外，其他的元素都有且只有一个前件和一个后件。图 1.4 给出了两个线性表的示例。

1	2	3	4	5	6	7
---	---	---	---	---	---	---

(a)

学号	姓名	外语	政治	C 语言
S061001	张三	85	54	92
S061002	李四	92	84	64
S061003	王五	87	74	73
S061004				
...				

(b)

图 1.4 表示例

在图 1-2 (a) 中开始元素为 1, 末尾元素为 7; 元素 3 的直接前驱为 2, 直接后继为 4。

图 1-2 (b) 中开始元素为第一条记录, 末尾元素为最后一条记录; 李四的成绩记录的前件为张三的成绩记录, 后件为王五的成绩记录。

非空线性表的特征:

▲ 只有一个根节点, 即 a_1 , 它无前件;

▲ 有且只有一个终端节点, 即 a_n , 它无后件;

▲ 除根节点和终端节点外, 其他所有结点有且只有一个前件, 也有且只有一个后件。

结点的个数 n 称为线性表的长度, 当 $n=0$ 时, 称为空表。

1.3.2 线性表的顺序存储结构

线性表的顺序存储指的是用一组地址连续的存储单元依次存放线性表中的各个元素。这种存储结构的特点是逻辑关系相邻的结点物理位置上也相邻, 其具有以下两个特点:

(1) 线性表中所有元素所占的存储空间是连续的;

(2) 线性表中各数据元素在存储空间中是按逻辑顺序依次存放的。

1.3.3 线性表的插入和删除运算

(1) 顺序表的插入运算: 在一般情况下, 要在第 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, 移动结束后, 第 i 个位置就被空出, 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。

(2) 顺序表的删除运算: 在一般情况下, 要删除第 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, 线性表的长度就减小了 1。

例题: 下列叙述中正确的是 ()。

A) 数据的逻辑结构与存储结构必定是一一对应的

B) 由于计算机存储空间是向量式的存储结构, 因此, 数据的存储结构一定是线性结构

C) 程序设计语言中的数据一般是顺序存储结构, 因此, 利用数组只能处理线性结构

D) 以上三种说法都不对。

答案为 D。本题为 2007 年 9 月份考试第 6 题。选项 A, 数据的逻辑结构与存储结构可以

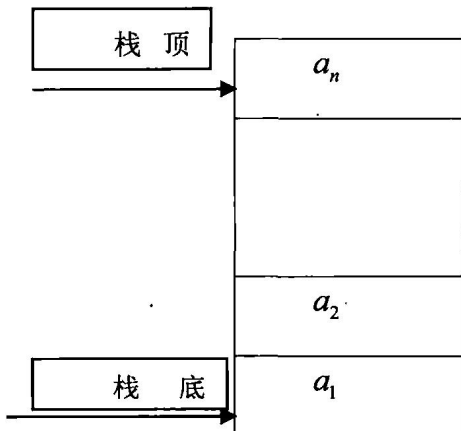
一一对应，也不一一对应；选项B，数据的存储结构有顺序、链接、索引等多种存储结构；选项C，数组也可以用来处理其他结构，只是这种方法不常用；故答案选D。

1.4 栈和队列

栈和队列是操作受限的线性表。

1.4.1 栈及其基本运算

栈是限定在一端进行插入与删除运算的线性表。

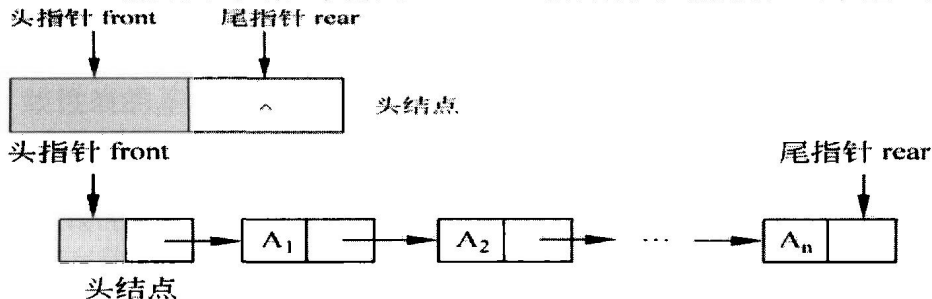


在栈中，允许插入与删除的一端称为栈顶，不允许插入与删除的另一端称为栈底。栈顶元素总是最后被插入的元素，栈底元素总是最先被插入的元素。即栈是按照“先进后出”或“后进先出”的原则组织数据的。

栈的基本运算：1) 插入元素称为入栈运算；2) 删除元素称为退栈运算或弹栈运算；3) 读栈顶元素是将栈顶元素赋给一个指定的变量，此时指针无变化。栈的存储方式和线性表类似，也有两种，即顺序栈和链式栈。

1.4.2 队列及其基本运算

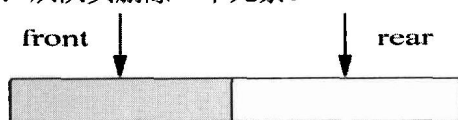
队列是指允许在一端（队尾）进行插入，而在另一端（队头）进行删除的线性表。尾指针（Rear）指向队尾元素，头指针（front）指向排头元素的前一个位置（队头）。



队列是“先进先出”或“后进后出”的线性表。

队列运算包括：

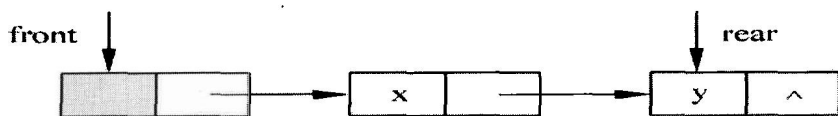
- (1) 入队运算：从队尾插入一个元素；
 (2) 退队运算：从队头删除一个元素。



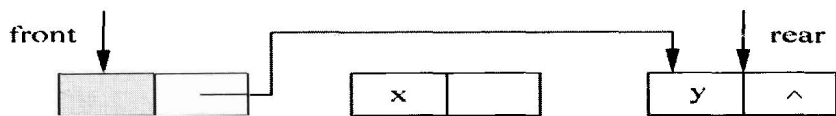
(a) 空队列



(b) 元素 x 入队列



(c) 元素 y 入队列



(d) 元素 x 出队列

循环队列及其运算：所谓循环队列，就是将队列存储空间的一个位置绕到第一个位置，形成逻辑上的环状空间，供队列循环使用。在循环队列中，用队尾指针 $rear$ 指向队列中的队尾元素，用排头指针 $front$ 指向排头元素的前一个位置，因此，从头指针 $front$ 指向的后一个位置直到队尾指针 $rear$ 指向的位置之间，所有的元素均为队列中的元素。

1.5 线性链表

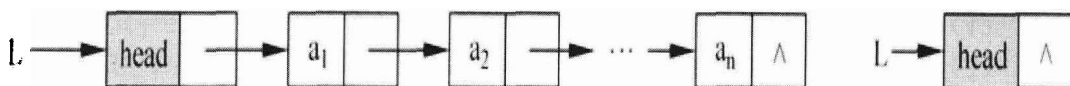
1.5.1 线性链表的基本概念

1. 链表

是一种常见的数据结构。与顺序表不同，它是动态的进行存储分配的一种结构，因而其长度不是固定。链表中的每一个元素称为“结点”，每个结点都由两个部分组成，即数据部分（数据域）和下一个结点的地址（指针域），从而可以表示数据元素的见后件关系。

2. 线性链表

线性表的链式存储结构称为线性链表，是一种物理存储单元上非连续、非顺序的存储结构，数据元素的逻辑顺序是通过链表中的指针链接来实现的。因此，在链式存储方式中，每个结点由两部分组成：一部分用于存放数据元素的值，称为数据域；另一部分用于存放指针，称为指针域，用于指向该结点的前一个或后一个结点（即前件或后件），如下图所示：



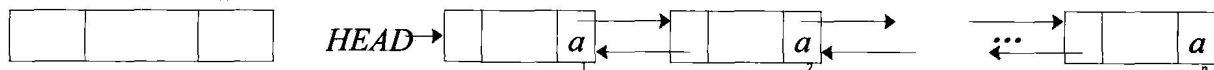
(a) 非空表

(b) 空表

线性链表分为单链表、双向链表和循环链表三种类型。

在单链表中，每一个结点只有一个指针域，由这个指针只能找到其后件结点，而不能找到其前件结点。因此，在某些应用中，对于线性链表中的每个结点设置两个指针，一个称为左指针，指向其前件结点；另一个称为右指针，指向其后件结点，这种链表称为双向链表，如下图所示：

左指针 数据域 右指针



(a) 结点结构

(b) 一个非空的双向链表示意图

1.5.2 线性链表的基本运算

(1) 在线性链表中包含指定元素的结点之前插入一个新元素。此时不需要移动数据元素，只需要修改相关结点指针即可。

(2) 在线性链表中删除包含指定元素的结点。此时也不需要移动数据元素，只需要修改相关结点指针即可。

(3) 将两个线性链表按要求合并成一个线性链表。

(4) 将一个线性链表按要求进行分解。

(5) 逆转线性链表。

(6) 复制线性链表。

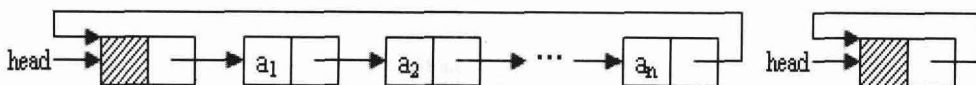
(7) 线性链表的排序。

(8) 线性链表的查找。

1.5.3 循环链表及其基本运算

与前面所讨论的单链表和双向链表相比，循环链表具有以下两个特点：1) 在链表中增加了一个表头结点，其数据域为任意或者根据需要来设置，指针域指向线性表的第一个元素的结点，而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。即在循环链表中，所有结点的指针构成了一个环状链。

下图 a 是一个非空的循环链表，图 b 是一个空的循环链表：



(a) 非空表

(b) 空表

循环链表的优点主要体现在两个方面：一是在循环链表中，只要指出表中任何一个结点的位置，就可以从它出发访问到表中其他所有的结点，而线性单链表做不到这一点；二

是由于在循环链表中设置了一个表头结点,在任何情况下,循环链表中至少有一个结点存在,从而使空表与非空表的运算统一。

优点:循环链表是在单链表的基础上增加了一个表头结点,其插入和删除运算与单链表相同。但它可以从任一结点出发来访问表中其他所有结点,并实现空表与非空表的运算的统一。

1.6 树与二叉树

树与二叉树是数据结构的重要部分,本节将对树和二叉树进行介绍,对其中的重点概念、重要术语举例说明,并对二叉树的基本性质、二叉树的遍历加以重点介绍。

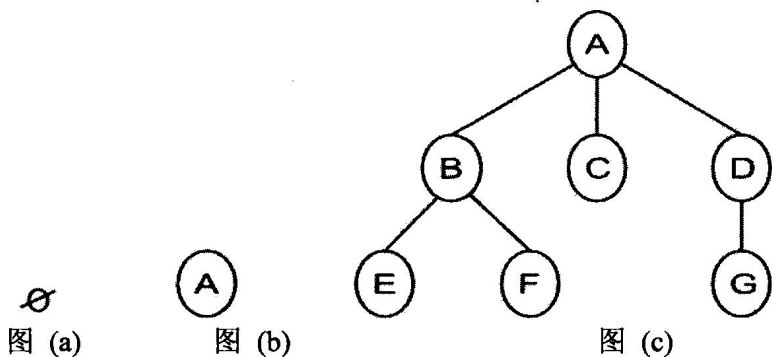
1.6.1 树的基本概念

树是一种非线性结构,是 n 个节点的有限集。当 $n=0$ 时,为空树;当 $n>0$ 时,为非空树。对于任意一棵非空树,都有以下两个特性:

(1) 有且仅有一个特定的称之为根 (root) 的结点,它只有直接后继,没有直接先驱。

(2) 当 $n>1$ 时,除根结点之外的其余结点可分为 $m(m>0)$ 个互不相交的有限集 $T_1, T_2, \dots, T_m$, 其中每个集合本身又是一棵树,并且称之为根的子树。每棵子树的根结点有且仅有一个直接先驱,但可以有 0 个或多个直接后继。

树的结构如树的示意图所示,它就像一棵倒长的树。图 (a) 是一棵空树,一个结点也没有。图 (b) 是一棵只有一个根结点的树。图 (c) 是一棵有 7 个结点的树。A 为根结点,其余分成 3 个互不相交的子集: $T_1 = \{B, E, F\}$, $T_2 = \{C\}$, $T_3 = \{D, G\}$, 它们都是 A 的子树。子树 T_1 的根结点是 B, 除 B 以外的求结点又组成 B 的两颗子树: $T_{11} = \{E\}$, $T_{12} = \{F\}$ 。其他的子树划分彼此类推。



树的示意图

下面关于树的一些基本概念非常重要,在后面的讲解中会经常出现。

◆**结点 (node):** 如图 (c) 所示的树中有 7 个结点,每个节点都包含数据项和若干个指向其子树的分支。

◆**结点的度:** 结点所拥有的子树棵数称为结点的度。如图 (c) 所示的树中,根结点 A 的度为 3。

◆**叶子结点:** 又称终端结点,是度为 0 的结点。如图 (c) 所示的树中, C、E、F、G 都是叶结点。

◆分支结点：一棵树中除叶结点外的其他结点都是分支结点，也可以称为非终端结点。如图(c)所示的树中，A、B、D都是分支结点。

◆子女结点：如果树中的某个结点有子树，则子树的根结点就是该结点的子女。如图(c)所示的树中，结点A有3分子女，即B、C和D；结点C没有子女。

◆双亲结点：如果某个结点有子女，则该结点就是其子女的双亲。如图(c)所示的树中，B、C、D有同一个双亲，就是A；根结点A没有双亲。

◆兄弟结点：双亲结点相同的各个结点互称为兄弟。如图(c)所示的树中，B、C、D互为兄弟，C、G不是兄弟。

◆祖先结点：从根结点到该结点所经分支上的所有结点。如图(c)所示的树中，E的祖先为A、B；C是祖先只有A。

◆子孙结点：以某结点为根的子树中的所有非根结点都是该结点的子孙。如图(c)所示的树中，A的子孙为B、C、D、E、F、G；B的子孙为E、F；C没有子孙。

◆结点的层次：根结点在第一曾，同一层上左右结点的子结点都在下一层。如图(c)所示的树中，跟结点A在第1曾，B、C、D都在第2曾，E、F、G都在第3曾。

◆树的深度：所处层次最大的哪个结点的层次称为树的深度。空树的高度为0，只有一个根结点的高度为1，图(c)所示的树的高度为3。

◆树的度：树中所有结点的度的最大值局势树的度。如图(c)所示的树的度为3（也就是A的度）。

1.6.2 二叉树及其基本性质

1. 二叉树的基本概念

二叉树是一种特殊的树形结构，它的每个结点最多有两棵树，也就是说，在二叉树中不存在度大于2的结点。除此之外，二叉树的子树还有左、右之分，也就是左、右子树不能互换。因此，二叉树有5种不同的形态，如下图所示。

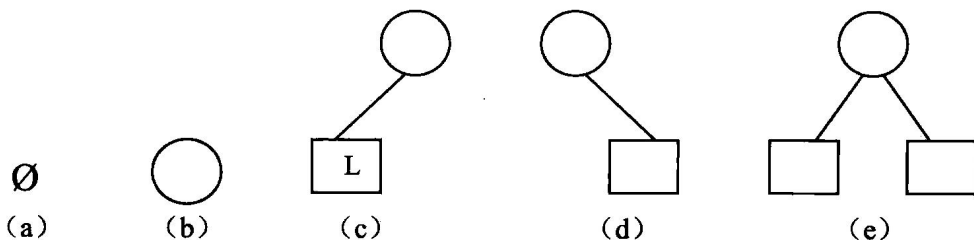


图 1-15 二叉树的 5 种不同形态 (L 表示左子树，R 表示右子树)

图(a)表示一棵空的二叉树；图(b)表示一棵只有根结点的二叉树，其左、右子树都为空；图(c)表示一棵根的右子树为空的二叉树；图(d)表示一棵根的左子树为空的二叉树；图(e)表示一棵根的左、右子树都不为空的二叉树。

2. 二叉树的基本性质

二叉树具有以下记过性质：

性质 1 在二叉树的第 k 层上，最多有 2^{k-1} ($k \geq 1$) 个结点。

性质 2 深度为 m 的二叉树最多有 $2^m - 1$ 个结点。

深度为 m 的二叉树表示该二叉树共有 m 层。根据性质 1，只要将第 1 层到第 m 层上的最大的结点数相加，就可以得到整个二叉树中结点的最大值，即 $2^{1-1} + 2^{2-1} + \dots + 2^{m-1} = 2^m - 1$ 。

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。此性质的推导比较复杂, 在此不做推导, 大家只要记住这条性质就行了。

性质4 具有 n 个结点的二叉树, 其深度不小于 $\lceil \log_2 n \rceil + 1$, 其中 $\lceil \log_2 n \rceil$ 为 $\log_2 n$ 的整数部分。

3. 满二叉树与完全二叉树

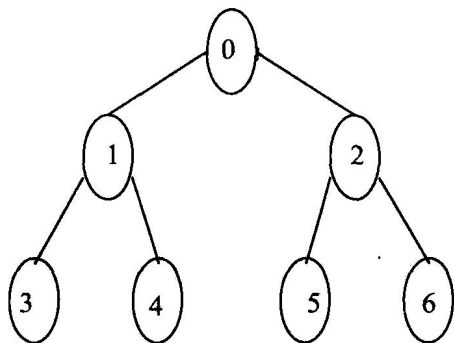
(1) 满二叉树

满二叉树是一棵深度为 m 且有 $2^m - 1$ 个结点的二叉树。该二叉树每一层上的所有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。如下图(a)所示就是一棵深度为3的满二叉树。

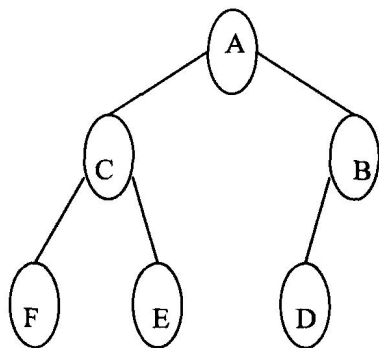
(2) 完全二叉树

完全二叉树除最后一层外, 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。更确切的说, 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 n , 且有 k 个结点的二叉树, 且仅当其每一个结点都与深度为 n 的满二

叉树中编号从1到 k 的结点一一对应时, 称之为完全二叉树。如下图(b)所示就是一棵深度为3的完全二叉树。



(a) 满二叉树 (深度为3)



(b) 完全二叉树 (深度为3)

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, 若其右分支下的子结点的最大层次为 p , 则其左右分支下的子孙结点的最大层次或为 p , 或为 $p+1$ 。

从上面满二叉树与完全二叉树的特点可以看出, 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。

完全二叉树具有下面两个性质:

性质1 具有 n 个结点的完全二叉树的深度为 $\lceil \log_2 n \rceil + 1$ 。

性质2 完全二叉树中度为1的结点数为0或1。

例题: 一棵二叉树中共有70个叶子结点与80个度为1的结点, 则该二叉树中的总结点数为()

- A) 219 B) 221 C) 229 D) 231

答案选A, 本题为2007年9月份考试第8题, 根据性质3, 度为0的结点(即叶子结点)总是比度为2的结点多1个, 故度为2的结点数为69个, 所以共有结点 $69 + 80 + 70 = 219$ 个, 故选A。