



国家级示范性高等院校精品规划教材

计算机公共基础

JISUANJI GONGGONG JICHU

高攀/主编



天津大学出版社
TIANJIN UNIVERSITY PRESS

国家级示范性高等院校精品规划教材

计算机公共基础

高 攀 / 主 编



天津大学出版社

TIANJIN UNIVERSITY PRESS

内 容 简 介

本书以教育部考试中心最新颁布的《全国计算机等级考试二级公共基础知识考试大纲(2009版)》为依据,以高教版教程为基础,结合编者多年从事培训辅导的实际经验编写而成。章节主体部分是知识点的讲解,精讲重点与难点;讲解过程中穿插真题和典型例题,并给出详细的解析;章节末安排适量习题并提供解答;书中附有数套笔试全真模拟试卷,供考生考前实战演练。

本书以全国计算机等级考试考生为主要读者对象,特别适合临考前冲刺复习使用,同时可以作为全国各类计算机等级考试培训班的教材以及大中专院校师生的参考书。

图书在版编目(CIP)数据

计算机公共基础 / 高攀主编. —天津: 天津大学出版社, 2010.7

国家级示范性高等院校精品规划教材

ISBN 978-7-5618-3569-2

I. ①计… II. ①高… III. ①电子计算机—高等学校—教材 IV. ①TP3

中国版本图书馆 CIP 数据核字(2010)第 130094 号

出版发行 天津大学出版社

出 版 人 杨欢

地 址 天津市卫津路 92 号天津大学内(邮编: 300072)

电 话 发行部: 022-27403647 邮购部: 022-27402742

网 址 www.tjup.com

印 刷 昌黎太阳红彩色印刷有限责任公司

经 销 全国各地新华书店

开 本 185mm×260mm

印 张 8.75

字 数 218 千

版 次 2010 年 7 月第 1 版

印 次 2010 年 7 月第 1 次

定 价 20.00 元

凡购本书, 如有缺页、倒页、脱页等质量问题, 请向我社发行部联系调换

版权所有 侵权必究

前 言

全国计算机等级考试 (National Computer Rank Examination, NCRE) 是由教育部考试中心主办, 面向社会, 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。其目的在于以考促学, 向社会推广和普及计算机知识, 也为用人单位录用和考核工作人员提供一个统一、客观、公正的标准。1994 年以来, 在我国众多的计算机考试中, 全国计算机等级考试已经是最具影响力的考试, 许多用人单位以全国计算机等级考试作为录用、上岗、晋升的必备条件之一。

易博 IT 培训中心长期从事全国计算机等级考试的培训工作, 培训通过率达到 85% 以上, 远远超过全国平均通过率, 受到广大学员的赞誉。为了让学员在较短时间内能快速提高应试能力, 顺利过关, 易博 IT 培训中心在全国范围内组织教学一线的有多年命题、阅卷及培训辅导经验的教师编写了全国计算机等级考试系列教材。

本书结合最新版考试大纲、教育部考试中心的指定教程, 以历年真题为基础, 以对考生进行综合指导、全面提高应试能力为原则, 深入研究最近几年的考试真题并结合编者实际经验编写而成。

本书按教育部考试中心指定教程的章节分类编排, 并按考试大纲的要求对近四年的试题进行详细的分析, 对相关知识点进行详尽的介绍, 使考生熟悉全国计算机等级考试二级公共基础知识的内容, 并抓住考试的重点与难点, 掌握考试中经常出现的题型和每种题型的解法, 同时也使考生熟悉专家们的出题思路、命题规律, 从而提高应试复习的效率, 做到有的放矢。

全书共分为四章, 第一章数据结构与算法、第二程序设计基础、第四章数据库设计基础由高攀老师编写, 第三章软件工程基础由李辉老师编写, 全书由高攀老师统稿。

本书不仅对准备参加计算机等级二级考试的读者大有裨益, 是应试者必备的自学和辅导材料, 而且对计算机爱好者、从事计算机教学的老师以及参加其他类似考试的读者也很有帮助。

在本书出版之际, 要特别感谢全国计算机等级考试的命题专家们。编者在本书中引用了历年考试的部分真题。此外, 本书在编写的过程中参考了许多相关的资料和书籍, 编者在此对这些参考文献的作者表示真诚的感谢。同时感谢天津大学出版社在本书出版过程中所给予的支持和帮助。

由于编者水平有限, 书中难免有不妥和错误之处, 编者诚恳期望各位专家和读者不吝赐教, 对此, 我们将深为感激。

编 者

2010 年 6 月

目 录

第一章 数据结构与算法	1
第一节 算法	2
第二节 数据结构的基本概念	5
第三节 线性表及其顺序存储结构	6
第四节 线性链表	9
第五节 栈及其基本运算	12
第六节 队列及其基本运算	14
第七节 树与二叉树	16
第八节 查找技术	23
第九节 排序技术	25
第十节 章节训练	29
第二章 程序设计基础	35
第一节 程序设计方法与风格	36
第二节 结构化程序设计	38
第三节 面向对象的程序设计	41
第四节 章节训练	45
第三章 软件工程基础	49
第一节 软件工程的基本概念	50
第二节 需求分析及其方法	57
第三节 软件设计及其方法	62
第四节 软件测试	72
第五节 程序的调试	79
第六节 章节训练	82
第四章 数据库设计基础	87
第一节 数据库系统的基本概念	88
第二节 数据模型	97
第三节 关系代数	108



第四节 数据库设计与管理	114
第五节 章节训练	123
附录一 考试大纲	127
附录二 全真模拟	129
参考文献	133

第一章

数据结构与算法

• 大纲要求 •

1. 算法的基本概念，算法复杂度（时间复杂度与空间复杂度）的概念和意义。
2. 数据结构的定义，数据的逻辑结构与存储结构，数据结构的图形表示，线性结构与非线性结构的概念。
3. 线性表的定义，线性表的顺序存储结构及其插入与删除运算。
4. 线性单链表、双向链表与循环链表的结构及其基本运算。
5. 栈和队列的定义，栈和队列的顺序存储结构及其基本运算。
6. 树的基本概念，二叉树的定义及其存储结构，二叉树的前序、中序和后序遍历。
7. 顺序查找与二分法查找算法，基本排序算法（插入类排序法、选择类排序法和交换类排序法）。

• 考试点拨 •

本章内容在笔试中会出现5~6个题目，是公共基础知识部分出题量比较多的一章，所占分值也比较大，约10分。本章中算法、数据结构及线性表等部分主要考查基本概念和一些性质，考生应在理解的基础上加强记忆。

重要的知识点包括栈和队列、线性链表、树和二叉树、查找和排序等。难点知识点包括二叉树遍历和排序等内容。考生应多做练习，以巩固这部分内容。

考生应注意，在掌握各种排序方法的比较次数和时间复杂度时，由于这两部分内容比较相似，记忆时可采取逆向思维，比如记住特殊的，那么基本上就能掌握全部内容了。



第一节 算法

重要考点

算法的基本概念

算法复杂度

一、算法的基本概念

算法是指解题方案的准确而完整的描述。算法不等于程序，也不等于计算方法。

1. 算法的基本特征

(1) 可行性

针对实际问题而设计的算法，执行后能够得到满意的结果。

(2) 确定性

算法中的每一个步骤都必须有明确的定义，不允许有模棱两可的解释和多义性。

(3) 有穷性

算法必须在有限时间内做完，即算法必须能在执行有限的步骤之后终止。

(4) 拥有足够的情报

一个算法执行的结果总是与输入的初始数据有关，不同的输入将会有不同的输出。当输入不够或输入错误时，算法本身无法执行或导致执行错误。所以，当算法拥有足够的情报来提供输入时，此算法才是有效的。

综上所述，算法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，同时是明确的，此顺序将在有限的次数后终止。

2. 算法的基本要素

(1) 算法中对数据的运算和操作

每个算法实际上是按解题要求从环境能进行的所有操作中选择合适的操作所组成的一组指令序列。

基本的运算和操作有以下四类。

1) 算术运算：主要包括“+”、“-”、“×”、“÷”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“>”、“<”、“=”、“≠”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

算法中各操作之间的执行顺序称为算法的控制结构。算法的基本控制结构主要分为顺

序、选择、循环三种。

3. 算法设计的基本方法

(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测哪些是需要的，哪些是不需要的。列举法常用于解决“是否存在”或“有多少种可能”等类型的问题。

(2) 归纳法

归纳法是从特殊到一般的抽象过程。通过分析少量的特殊情况，找出一般的关系。归纳法比列举法更能反映问题的本质，并且可以解决无限列举量的情况。但是归纳法不容易实现。

(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。其中初始条件或是问题本身已经给定，或是通过对问题的分析与化简而确定。递推法本质上属于归纳法。

(4) 递归法

递归法分为直接递归与间接递归两种。如果一个算法 A 显式地调用自己则称为直接递归；如果算法 A 调用另一个算法 B，而算法 B 又调用算法 A，则称为间接递归。

(5) 减半递推法

实际问题的复杂程度往往与问题的规模有着密切的联系，因此利用分治法解决这类实际问题是有有效的。工程上常用的分治法是减半递推法。

(6) 回溯法

对需要解决的问题进行一条线索逐步试探，如果试探成功，就找到了解决的方法；如果试探失败，就逐步回退，然后换另一条线索进行试探。

二、算法复杂度

算法的复杂度包括时间复杂度和空间复杂度。

1. 算法的时间复杂度

通常用算法在执行过程中所需基本运算的执行次数来度量算法的工作量。基本运算反映了算法运算的主要特征，同一个算法使用不同的语言实现，或者用不同的编译程序进行编译，或者在不同的计算机上运行，效率均不同。这表明使用绝对的时间单位衡量算法的效率是不合适的，因此用基本运算的次数来度量算法工作量是客观的也是实际可行的，有利于比较同一问题的几种算法的优劣。

分析算法的工作量通过采用以下两种方法来分析。

(1) 平均性态 (Average Behavior)



所谓平均性态分析,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) 最坏情况复杂性 (Worst-case Complexity)

所谓最坏情况分析,是指在规模为 n 时,算法所执行的基本运算的最大次数。

2. 算法的空间复杂度

算法的空间复杂度是指执行算法所需要的内存空间。一个算法所占用的存储空间包括:①算法程序所占的空间;②输入的初始数据所占的存储空间;③算法执行过程中所需的额外空间。

【例 1-1】算法的时间复杂度是指_____。

- A. 执行算法程序所需要的时间
- B. 算法程序的长度
- C. 算法执行过程中所需要的基本运算次数
- D. 算法程序中的指令条数

例题分析 算法的时间复杂度是指执行算法所需要的计算工作量。为了能够比较客观地反映出一个算法的效率,在度量一个算法的工作量时,不仅应该与所使用的计算机、程序设计语言以及程序编制者无关,而且还应该与算法实现过程中的许多细节无关。为此,可以用算法在执行过程中所需基本运算的执行次数来度量算法的工作量。

答案: C

【例 1-2】算法的空间复杂度是指_____。

- A. 算法程序的长度
- B. 算法程序中的指令条数
- C. 算法程序所占的存储空间
- D. 算法执行过程中所需要的存储空间

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

答案: D

【例 1-3】算法是对问题求解过程的一类精确描述,算法中描述的操作都可以通过已经实现的基本操作在限定时间内执行有限次来完成的,这句话说明算法应具有_____特性。

例题分析 算法应满足有穷性、确定性、可行性、拥有足够的情报等基本特征,而有穷性即为在有限时间内完成算法。

答案: 有穷性

第二节 数据结构的基本概念

重要考点

逻辑结构和存储结构

线性结构和非线性结构

一、数据结构

数据结构是相互之间存在一种或多种特定关系的数据元素的集合。在任何数据处理问题中，数据元素都不是孤立存在的，在它们之间存在着某种关系。

数据结构作为计算机的一门学科，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所固有的逻辑关系。
- 2) 数据的存储结构：在对数据进行处理时，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。

1. 数据的逻辑结构

数据的逻辑结构，是描述数据元素之间逻辑关系的数据结构。数据的逻辑结构由某一数据对象及该对象中所有数据成员之间的关系（前后件关系）组成。即一个数据结构可以表示成

$$\text{Data_Structure} = (D, R)$$

其中， D 是数据元素的集合， R 是 D 上的关系，它反映了 D 中各数据元素之间的前后件关系。其中，数据元素之间的前后数据关系是指它们的逻辑关系，而与它们在计算机中的存储位置无关。

2. 数据的存储结构

一个数据结构中的各元素在计算机存储空间中的位置关系与逻辑关系是有可能不同的。数据的逻辑结构在计算机中的存储方式称为数据的存储结构（也称数据的物理结构）。它包括数据元素的存储方式和关系的存储方式。

通常，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。采用不同的存储结构，其数据处理的效率是不同的。

二、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相关联的复杂程度，通常将数据结构分为两大类型：线性结构和非线性结构。

如果一个非空数据结构满足两个条件：① 有且只有一个根节点；② 每一个节点最多有



一个前件，也最多有一个后件，则称该数据结构为线性结构，线性结构又称为线性表。例如，线性表、栈、队列和线性链表都是线性结构。

如果一个数据结构不是线性结构，称之为非线性结构。例如树、二叉树和图都是非线性结构。

线性结构与非线性结构都可以是空的数据结构。对于空的数据结构，如果对该数据结构的运算是按线性结构的规则来处理的，则属于线性结构，否则属于非线性结构。

【例 1-4】数据的存储结构是指_____。

- A. 存储在外存中的数据 B. 数据所占的存储空间量
C. 数据在计算机中的顺序存储方式 D. 数据的逻辑结构在计算机中的表示

例题分析 数据的逻辑结构在计算机存储空间中的存放形式称为数据的存储结构（也称为数据的物理结构）。通常有两类存储结构：一类是顺序存储结构；另一类是链式存储结构。

答案：D

【例 1-5】数据结构中，与所使用的计算机无关的是数据的_____。

- A. 存储结构 B. 物理结构 C. 逻辑结构 D. 物理和存储结构

例题分析 数据的逻辑结构只抽象地反映数据元素之间的逻辑关系，即数据元素之间的前后件关系，而不取决于在计算机中的存储表示形式。

答案：C

第三节 线性表及其顺序存储结构

重要考点

线性表的基本概念

顺序表的顺序存储结构

顺序表的插入、删除运算

一、线性表的基本概念

线性表是最简单且最常用的一种线性数据结构。

一个线性表是 n 个数据元素的有限序列，可以表示为 $(a_1, a_2, \dots, a_i, \dots, a_n)$ ，其中 a_i ($i=1, 2, \dots, n$) 是属于数据对象的元素，通常也称其为线性表中的一个节点。

线性表可以是一个空表，非空线性表具有以下一些特征。

- ① 有且只有一个根节点，且无前件。
- ② 有且只有一个终端节点，且无后件。

③ 除根节点和终端节点外，其他所有节点有且只有一个前件和一个后件。

二、线性表的顺序存储结构

线性表的顺序存储结构指的是用一组地址连续的存储单元依次存放线性表中的数据元素。线性表的顺序存储结构具备如下两个基本特征：

- ① 线性表中的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。

因此，在线性表的顺序存储结构中，具有前后件关系的两个元素在存储空间是紧邻的，且前件元素一定存储在后件元素的前面。

假设线性表中的第一个数据元素的存储地址（指第一个字节的地址，即首地址）为 $ADR(a_1)$ ，每个数据元素占 k 个字节，则线性表中第 i 个元素在计算机存储空间的存储地址为

$$ADR(a_i) = ADR(a_1) + (i-1) * k$$

在程序语言中，通常定义一个一维数组表示线性表的顺序存储空间。

三、线性顺序表的插入运算

图 1-1 (a) 为一个长度为 6 的线性表顺序存储在长度为 7 的存储空间中。现在要求在第 5 个元素（即 78）之前插入一个新元素 25。

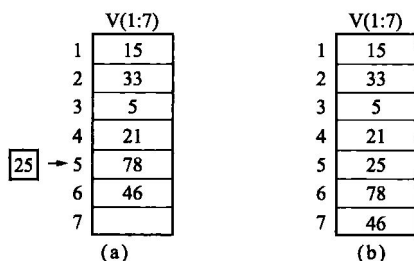


图 1-1 线性表在顺序存储结构下的插入
(a) 插入前；(b) 插入后

具体操作步骤为：首先从最后一个元素开始直到第 5 个元素，将其中的每一个元素均依次往后移动一个位置，然后将新元素 25 插入到第 5 个位置；插入一个新元素后，线性表的长度变成了 7，如图 1-1 (b) 所示。这时，为线性表开辟的存储空间已经满了，如果再要插入，则会造成称为“上溢”的错误。

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 $n-i+1$ 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。



四、线性顺序表的删除运算

图 1-2 (a) 为一个长度为 6 的线性表顺序存储在长度为 7 的存储空间中。现在要求删除线性表中的第 3 个元素 (即删除元素 5)。

V(1:7)		V(1:7)	
1	15	1	15
2	33	2	33
3	5	3	21
4	21	4	78
5	78	5	46
6	46	6	
7		7	

(a)

(b)

图 1-2 线性表在顺序存储结构下的删除

(a) 删除前; (b) 删除后

具体操作步骤为: 从第 4 个元素开始直到最后一个元素, 将其中的每一个元素均依次往前移动一个位置。此时, 线性表的长度变成了 5, 如图 1-2 (b) 所示。

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。

在顺序表中插入或删除一个数据元素, 平均约需移动表中一半元素。这个数目在线性表的长度较大时是很可观的。这个缺陷完全是由于顺序存储要求线性表的元素依次紧挨存放造成的。因此, 顺序存储表仅适用于不常进行插入和删除操作、表中元素相对稳定的线性表。

【例 1-6】在长度为 n 的顺序表的第 i ($1 \leq i \leq n+1$) 个位置上插入一个元素, 元素的移动次数为_____。

A. $n-i+1$

B. $n-i$

C. i

D. $i-1$

例题分析 设插入位置为 i , 需要将第 i 个位置起后面所有的 (共 $n-i+1$ 个) 元素向后移动, 由此可知, 在第 i ($1 \leq i \leq n+1$) 个位置上插入一个元素, 元素的移动次数为 $n-i+1$ 。

答案: A

【例 1-7】以下描述中, 不是线性表的顺序存储结构的特征的是_____。

A. 不便于插入和删除

B. 需要连续的存储空间

C. 可随机访问

D. 需另外开辟空间来保存元素之间的关系

例题分析 线性表的顺序存储是用一片连续空间来存放数据元素, 其特点是逻辑上相邻的元素在物理位置上也相邻, 数据元素之间逻辑上的先后关系自动隐含在物理位置的相邻之中, 因此不需另外开辟空间来保存元素之间的关系。

答案: D

【例 1-8】对于长度为 n 的顺序表进行插入运算和删除运算, 可能会引起元素的移动。在等概率情况下, 插入操作平均需要移动 ① 个元素; 删除操作平均需要移动 ② 个元素。

例题分析 将各种位置上插入和删除的移动次数相加再除以元素总个数即可。

答案: ① $n/2$; ② $(n-1)/2$

第四节 线性链表

重要考点

链表的存储结构

线性链表的基本操作

双向链表和循环链表

一、线性链表的基本概念

线性表的顺序存储结构具有简单、操作方便等优点，但在对其做插入或删除操作时，需要移动大量的元素。因此，对于大的线性表，特别是元素变动频繁的大线性表不宜采用顺序存储结构，而是通常采用链式存储结构。在链式存储结构中，存储数据结构的存储空间可以不连续，各数据节点的存储顺序与数据元素之间的逻辑关系可以不一致。

链式存储结构中的每一个数据节点对应于一个存储单元（若干个字节），这种存储单元称为存储节点，简称节点。在链式存储方式中，要求每个节点由两部分组成：一部分用于存放数据元素值，称为数据域；另一部分用于存放指针，称为指针域。其中指针用于指向该节点的前一个或后一个节点，从而可以表示数据元素之间的前后件关系。

线性表的链式存储结构称为线性链表。线性链表中存储节点的结构如图 1-3 所示，线性链表的逻辑结构如图 1-4 所示。

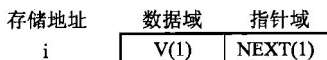


图 1-3 线性链表的一个存储节点

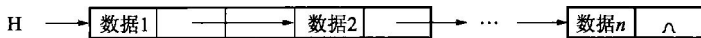


图 1-4 线性链表的逻辑结构

在线性链表中，用一个专门的指针 HEAD（称为头指针）指向线性链表中第一个数据元素的节点（即存放线性表中第一个数据元素的存储节点的序号）。从头指针开始，沿着线性链表各节点的指针可以扫描到链表中的所有节点。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节点的指针域为空（用 NULL 或 0 表示），表示链表终止。当头指针 HEAD = NULL（或 0）时称为空表。

二、线性链表的物理状态和逻辑状态

假设某线性表为 (b_1, b_2, b_3, b_4) ，存储空间具有 7 个存储节点，该线性表的存储情况如图 1-5 (a) 所示，图 1-5 (b) 所示的是线性链表的逻辑状态，其中每一个节点上面的数字



表示该节点的存储地址。

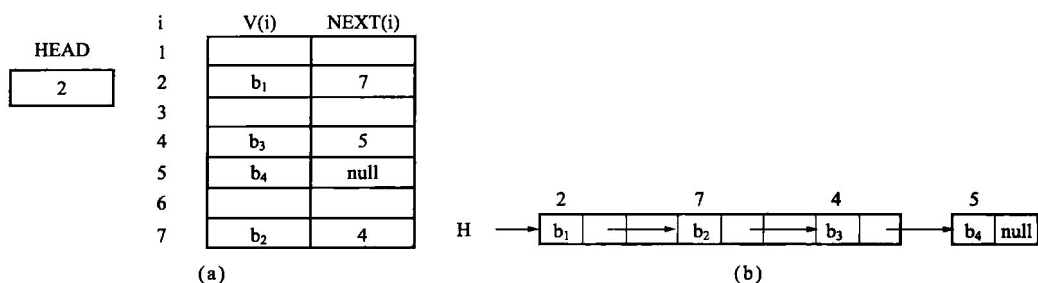


图 1-5 线性链表的物理状态和逻辑状态示例

(a) 线性链表的物理状态; (b) 线性链表的逻辑状态

三、双向链表

在一些应用中, 对线性链表中的每个节点需设置两个指针: 一个称为左指针, 用以指向其直接前件元素; 另一个称为右指针, 用以指向其直接后件元素。这样的线性链表称为双向链表, 其逻辑状态如图 1-6 所示。



图 1-6 具有三个节点的双向链表

四、线性链表的基本操作

(1) 在线性链表中查找指定的元素

在非空线性链表中寻找包含指定元素值 x 的前一个节点 n 的操作过程如下。

从头指针指向的节点开始向后沿指针进行扫描, 直到后面已经没有节点或下一个节点的数据域为 x 为止。因此, 由这种方法找到的节点 n 有两种可能: 当线性链表中存在包含元素 x 的节点时, 则找到的 n 为首次发现的包含元素 x 的前一个节点序号; 当线性链表中不存在包含元素 x 的节点时, 则找到的 n 为线性链表中的最后一个节点序号。

(2) 线性链表的插入

线性链表的插入操作是指在线性链表中的指定位置上插入一个新的元素。为了要在线性链表中插入一个新元素, 首先要为该元素申请一个新节点, 以存储该元素的值。然后将存放新元素值的节点链接到线性链表中指定的位置。

(3) 线性链表的删除

线性链表的删除是指在线性链表中删除包含指定元素的节点。

为了在线性链表中删除包含指定元素的节点, 首先要在线性链表中找到这个节点, 然后将要删除的节点释放, 以便于以后再次利用。

五、循环链表及其基本操作

循环链表的结构与前面所讨论的线性链表相比, 具有以下两个特点。

1) 在循环链表中增加了一个表头节点, 表头节点的数据域可以是任意值, 也可以根据需要来设置, 指针域指向线性表的第一个元素的节点。循环链表的头指针指向表头节点。

2) 循环链表中最后一个节点的指针域不为空, 而是指向表头节点。从而在循环链表中, 所有节点的指针构成了一个环。

因此在对循环链表进行插入与删除的操作时, 无须对空表和对第一个节点的处理单独考虑, 使空表与非空表的操作得到了统一, 从而克服了线性链表的这两大缺点。

图 1-7 是循环链表的示意图。其中图 1-7 (a) 是一个非空的循环链表, 图 1-7 (b) 是一个空的循环链表。由于在循环链表中设置了一个表头节点, 因此, 在任何情况下, 循环链表中至少有一个节点存在, 从而使对空链表与非空链表的操作统一起来。

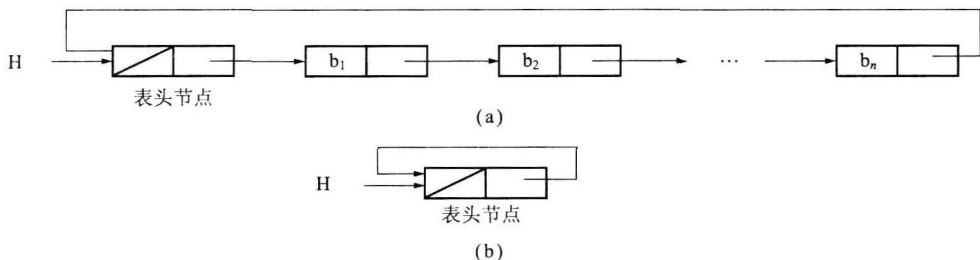


图 1-7 循环链表示意图

(a) 非空循环链表; (b) 空循环链表

【例 1-9】下列对于线性链表的描述中正确的是_____。

- A. 存储空间不一定连续, 且各元素的存储顺序是任意的
- B. 存储空间不一定连续, 且前件元素一定存储在后件元素的前面
- C. 存储空间必须连续, 且各前件元素一定存储在后件元素的前面
- D. 存储空间必须连续, 且各元素的存储顺序是任意的

例题分析 线性表的链式存储结构中的节点空间是动态生成的, 它们在内存中的地址可以是连续的, 也可以是不连续的, 即逻辑上相邻, 存储(物理)上不一定相邻。

答案: A

【例 1-10】下列描述的不是链表的优点是_____。

- A. 逻辑上相邻的节点物理上不必邻接
- B. 插入、删除运算操作方便, 不必移动节点
- C. 所需存储空间比线性表节省
- D. 无须事先估计存储空间的大小

例题分析 线性表的链式存储是用一组任意的存储空间来存放数据元素, 链表节点空间是动态生成的, 无须事先估计存储空间的大小。链表逻辑上相邻的元素在物理位置上不一定相邻, 因此需要另外开辟空间来保存元素之间的关系, 花费的存储空间较顺序存储多。在链表中插入或删除节点, 只需修改指针, 不需要移动元素。

答案: C