



高职高专计算机系列规划教材

数据结构 (第二版)

包振宇 孙 干 陈 勇 编著

DATA
STRUCTURE



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高职高专计算机系列规划教材

数 据 结 构

(第二版)

包振宇 孙 干 陈 勇 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

教材内容以“实践应用”为主体,理论以“够用”为尺度,理论与实验相结合。其内容分为两部分,包括理论知识部分与实验部分。本教材有如下特点:

(1) 所有例题都包括示意图、分析、流程图和程序代码四个部分,思路清晰,层次鲜明,能逐步培养和提高学生分析问题和解决问题的能力。每章均配有适量习题和实验,具有很强的针对性和可操作性。

(2) 以全国计算机程序员考试大纲为基准,涉及考试的章节选用部分历年试题作为示例,以加强学生对所学内容的进一步理解、巩固和应用。

(3) 书中的算法和实验程序用标准 C 语言函数实现,可直接在 Turbo C 或 Visual C++ 6.0 环境下运行。

本书适合于高职高专院校计算机类专业的学生。

图书在版编目(CIP)数据

数据结构/包振宇,孙干,陈勇编著.—2版.—北京:

中国铁道出版社,2009.5

(高职高专计算机系列规划教材)

ISBN 978-7-113-10011-7

I. 数… II. ①包… ②孙… ③陈… III. 数据结构—高等

学校:技术学校—教材 IV. TP311.12

中国版本图书馆 CIP 数据核字(2009)第 72834 号

书 名: 数据结构(第二版)

作 者: 包振宇 孙 干 陈 勇 编著

策划编辑: 严晓舟 秦绪好

责任编辑: 翟玉峰

编辑助理: 邱雪姣

封面设计: 付 巍

编辑部电话: (010) 63583215

封面制作: 白 雪

责任印制: 李 佳

出版发行: 中国铁道出版社(北京市宣武区右安门西街8号 邮政编码: 100054)

印 刷: 三河市华业印装厂

版 次: 2005年8月第1版 2009年6月第2版 2009年6月第6次印刷

开 本: 787mm×1092mm 1/16 印张: 12 字数: 280千

印 数: 5 000 册

书 号: ISBN 978-7-113-10011-7/TP·3272

定 价: 20.00 元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签,无标签者不得销售

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

第二版前言

“数据结构”是计算机程序设计的重要基础，也是计算机科学与技术等电子信息类相关专业的一门重要基础课程。它对程序设计思想的建立和提升有着重要的作用，既为后续的计算机课程奠定了一个较为扎实的基础，又可提高学生分析问题和解决问题的能力。

本书作为 2005 年版《数据结构》的第二版，保持了前一版的基本框架，概念清楚、论述充分、面向应用，进一步完善了算法与数据结构体系的内容，每章都增加了“典型例题”一节，并对实验部分进行调整和补充。教材主要面向高职高专院校计算机科学与技术等电子信息类专业的学生。教材内容以“实践应用”为主体，理论以“够用”为尺度。全书共分理论知识和实验指导两个部分，理论知识部分又分为八章：第 1 章，绪论；第 2 章，线性表；第 3 章，栈和队列；第 4 章，字符串、数组和广义表；第 5 章，树；第 6 章，图；第 7 章，查找；第 8 章，排序。实验部分精选 18 个典型实例以帮助学生巩固所学知识。

本教材有如下特点：

(1) 所有例题都包括示意图、分析、流程图和程序代码四个部分，思路清晰，层次鲜明，能逐步培养和提高学生分析问题和解决问题的能力。本书每章均配有适量习题和实验，具有很强的针对性和可操作性。

(2) 以全国计算机程序员考试大纲为基准，涉及考试的章节选用部分历年试题作为示例，加强学生对所学内容的进一步理解、巩固和应用。

(3) 各章都配有“典型例题”一节，列举分析了很多实用的例子，有助于学生加深对基础理论知识的理解和实际应用能力的培养。

(4) 本书的算法和实验程序用标准 C 语言函数实现，可直接在 Turbo C 或 Visual C++ 6.0 环境下运行。

(5) 为配合本教材的教学，还编制了多媒体课件，以加深对基本概念的理解。课件和习题答案，可通过 E-mail (laobaomd@163.com) 向包振宇索取。

建议本教材的理论教学时数为 40 学时，实验教学时数为 30 学时。

本教材由包振宇、孙干、陈勇编著。其中，理论部分的第 1~5 章和实验部分由包振宇编写，理论部分的第 6、7 章和习题部分由孙干编写，理论部分的第 8 章由陈勇编写，最后由包振宇完成全书统稿。周永华、朱其俊参与了编写大纲的讨论并编写了初稿的部分内容，陈刚、张琴、刘呈真、杨健等参与了全书的校订工作，刘爱华、吴亮等在使用本书的过程中提出了一些修改建议，使得本书更加完善。

本书在编写和修订过程中，得到了许多专家和众多院校数据结构任课教师的大力支持和帮助，他们提出了许多中肯的意见和很好的建议，对本书的编写及修订起到了很大的指导作用。编者对此表示衷心的感谢。

由于“数据结构”依然是一门年轻的学科，内容还在不断地变化与更新，加之作者水平有限以及时间仓促等原因，书中难免有疏漏与不足之处，敬请广大读者批评指正。

编 者

2009 年 4 月

第一版前言

“数据结构”是计算机程序设计的重要基础，也是计算机相关专业的一门重要基础课程和核心课程。它对程序设计思想的建立、提升有着重要的作用，既为后续的计算机课程奠定了一个较为扎实的基础，又可提高学生分析问题和解决问题的能力。

本教材主要面向高职高专院校计算机类专业的学生。教材内容以“实践应用”为主体，理论以“够用”为尺度。全书共分理论知识和实验指导两个部分，理论知识部分又分为 8 章：第 1 章绪论；第 2 章线性表；第 3 章栈和队列；第 4 章字符串、数组和广义表；第 5 章树；第 6 章图；第 7 章查找；第 8 章排序。实验部分精选 15 个典型实例帮助学生巩固所学知识。

本教材有如下特点：

(1) 所有例题都分为：示意图、分析、流程图和函数 4 个部分，思路清晰，层次鲜明，能逐步培养和提高学生分析问题和解决问题的能力。

(2) 以全国计算机程序员考试大纲为基准，涉及考试的章节选用部分历年试题举例，加强学生对所学内容的进一步理解、巩固和应用。

(3) 本书的算法用标准 C 语言的函数来实现，加上主函数后可直接在 Turbo C 或 Visual C++ 6.0 环境下运行。

(4) 本书每章配有适量习题和实验，具有很强的针对性和可操作性。实验程序可直接在 Turbo C 或 Visual C++ 6.0 环境下运行。

建议本教材理论教学时数为 40 学时左右，实验教学时数为 30 学时左右。

本教材理论部分的第 1、2、3、4、5、8 章和实验部分由包振宇编写，理论部分的第 6、7 章和习题部分由孙干老师编写，最后由包振宇完成全书统稿。

由于时间仓促和编者水平有限，书中难免存在不妥或疏漏之处，敬请广大读者与专家批评指正。

编者

2005 年 7 月

目 录

第一部分 理论知识

第 1 章 绪论	1
1.1 数据结构与算法	1
1.1.1 数据结构的基本概念	1
1.1.2 算法的概念和特性	2
1.2 算法的描述和分析	2
1.2.1 算法的描述	2
1.2.2 算法的分析	2
1.3 典型例题	3
习题 1	4
第 2 章 线性表	7
2.1 线性表的逻辑结构	7
2.2 线性表的顺序存储结构	7
2.2.1 顺序分配	7
2.2.2 线性表的操作	8
2.3 线性表的链式存储结构	12
2.3.1 线性链表的实现	13
2.3.2 线性链表的运算	13
2.3.3 循环链表	18
2.4 典型例题	22
习题 2	25
第 3 章 栈和队列	29
3.1 堆栈	29
3.1.1 堆栈的定义和基本操作	29
3.1.2 顺序存储栈	29
3.1.3 链式存储栈	30
3.2 队列	31
3.2.1 顺序存储队列	32
3.2.2 链式存储队列	33
3.3 典型例题	34
习题 3	36

第 4 章	字符串、数组和广义表	39
4.1	字符串基本概念	39
4.2	字符串的存储结构	39
4.2.1	串的顺序存储结构	39
4.2.2	串的链式存储结构	40
4.3	字符串的模式匹配	40
4.3.1	模式匹配的 BF 算法	40
4.3.2	模式匹配的 KMP 算法	42
4.4	数组的基本概念	47
4.5	矩阵的压缩存储	47
4.5.1	特殊矩阵的压缩	47
4.5.2	稀疏矩阵	48
4.6	广义表	55
4.6.1	广义表的存储结构	56
4.6.2	综合举例	57
4.7	典型例题	58
习题 4		61
第 5 章	树	63
5.1	树的定义和术语	63
5.1.1	树的定义	63
5.1.2	树的基本术语	63
5.2	二叉树	64
5.2.1	二叉树的定义和性质	64
5.2.2	二叉树的存储结构	65
5.3	遍历二叉树	67
5.3.1	遍历二叉树的方法	67
5.3.2	遍历二叉树的函数	68
5.4	线索二叉树	69
5.5	树和森林	71
5.5.1	树的存储结构	71
5.5.2	树与二叉树的转换 (通过二叉链表存储联系)	73
5.6	树的应用	74
5.6.1	二叉排序树	74
5.6.2	哈夫曼树	76
5.7	典型例题	80
习题 5		82
第 6 章	图	87
6.1	图的基本概念	87

6.1.1	图的定义	87
6.1.2	图的相关术语	88
6.2	图的存储结构	90
6.2.1	邻接矩阵	90
6.2.2	邻接表	91
6.3	图的遍历	92
6.3.1	深度优先搜索 (DFS)	92
6.3.2	广度优先搜索 (BFS)	93
6.4	最小代价生成树	94
6.4.1	最小代价生成树的概念	94
6.4.2	构造最小生成树的 PRIM 算法	95
6.5	最短路径	96
6.5.1	从某个顶点到其他顶点的最短路径	96
6.5.2	每一对顶点间的最短路径	97
6.6	拓扑排序	98
6.7	典型例题	100
	习题 6	101
第 7 章	查找	106
7.1	线性表的查找	106
7.1.1	顺序存储线性表的查找	106
7.1.2	分块查找	109
7.1.3	链式存储线性表查找	109
7.2	树的查找	110
7.2.1	二叉树查找	110
7.2.2	平衡二叉树	112
7.2.3	B 树	114
7.3	哈希表及其查找	115
7.3.1	哈希表	115
7.3.2	常见的散列函数	115
7.3.3	解决冲突的方法	116
7.4	典型例题	117
	习题 7	118
第 8 章	排序	120
8.1	选择排序	120
8.2	直接插入排序	124
8.2.1	顺序存储线性表的直接插入排序	124
8.2.2	链式存储线性表的直接插入排序	125

8.3	冒泡排序	129
8.3.1	顺序存储线性表的冒泡排序	129
8.3.2	链式存储线性表的冒泡排序	130
8.4	希尔排序	131
8.5	堆排序	132
8.6	快速排序	134
8.7	合并排序	136
8.8	典型例题	137
	习题 8	139

第二部分 实验部分

实验一	时间复杂度的计算	142
实验二	顺序存储线性表的操作 (1)	143
实验三	顺序存储线性表的操作 (2)	145
实验四	链式存储线性表的操作 (1)	147
实验五	链式存储线性表的操作 (2)	150
实验六	链式存储线性表的操作 (3)	151
实验七	顺序栈的操作	153
实验八	顺序存储队列的进队列和出队列操作	155
实验九	字符串的操作	157
实验十	数组的操作	159
实验十一	广义表的操作	160
实验十二	树的操作	166
实验十三	图的操作	169
实验十四	二分法查找的操作	174
实验十五	插入排序的操作	176
实验十六	选择排序的操作	178
实验十七	快速排序的操作	180
实验十八	冒泡排序和希尔排序的操作	182
参考文献		184

第1章 绪论

“数据结构”形成和发展的背景：自 1946 年第一台计算机问世以来，计算机已深入到人类社会的各个领域，计算机的应用已不再局限于科学计算，而更多地用于控制、管理及信息处理等非数值计算的工作。计算机加工处理的对象由纯粹的数值发展到字符、表格、图像、声音等各种具有一定结构的数据，这就给程序设计带来一些新的问题。为编写出一个“好”的程序，必须分析待处理的对象的特性以及各处理对象之间存在的关系。这就是“数据结构”这门学科形成和发展的背景。

1.1 数据结构与算法

1.1.1 数据结构的基本概念

1. 数据

对现实世界的事物采用计算机能识别、存储和处理的形式所进行的描述。简言之，数据是计算机程序能加工和处理的对象，也就是计算机化的信息。

2. 数据元素

数据的基本单位，又称为结点。在计算机程序中通常作为一个整体进行考虑和处理。有时一个数据元素可由若干个数据项组成。如一本书的书目信息为一个数据元素，而书目信息中的每一项（如书名、作者名等）为一个数据项。数据项是数据不可分割的最小单元。

3. 数据对象

数据对象是性质相同的数据元素的集合，是数据的一个子集。

4. 数据结构

(1) 相互之间存在一种或多种特定关系的数据元素的集合。数据结构是指数据对象及其相互关系和构造方法。一个数据结构 DS 可以被形象地用一个二元组表示为： $DS=(D,R)$ 。其中， D 是数据结构中的数据（称为结点）的非空集， R 是定义在 D 上的关系的非空有限集合。结构是指结点之间的关系，数据结构就是结点的有限集合和关系的有限集合。

(2) 数据结构中，结点及结点间的相互关系是数据的逻辑结构。数据结构在计算机中的存储内容，一般包括结点的值和结点间的关系，数据结构的存储形式是数据的存储结构（或物理结构）。

(3) 数据结构按逻辑关系的不同分为线性结构和非线性结构两大类，其中非线性结构又分为树形结构和图状结构，而树形结构又可分为树结构和二叉树结构。

(4) 在计算机中，表示信息的最小单位是二进制数的一位，称为位（bit）。用一位或

若干位组合起来形成的一个位串表示一个数据元素或结点。元素或结点可看成是数据元素在计算机中的映像,有顺序映像和非顺序映像之分。相应地,存储结构可分为顺序存储结构和链式存储结构。顺序映像的特点是借助元素在存储器中的相对位置来表示数据元素之间的逻辑关系。非顺序映像的特点是借助指示元素存储地址的指针表示数据元素之间的逻辑关系。

5. 数据类型

指某种程序设计语言所允许使用的变量类型。

1.1.2 算法的概念和特性

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,其中每一条指令表示一个或多个操作。在计算机科学中,算法则是描述计算机解决特定问题的操作过程。

算法的特性是有穷性、确定性、可行性、输入和输出。

- (1) 有穷性: 求解问题的步骤是有限的,且每一步都可在有限时间内完成。
- (2) 确定性: 在任何情况下,对同一问题的求解结果必须是唯一的。
- (3) 可行性: 算法的实现必须在法律上、经济上、技术上都可行。
- (4) 输入: 如果没有输入,对问题的求解方法仅能适用于这一个特定问题。
- (5) 输出: 如果没有输出,就看不到结果,对问题的求解将失去任何意义。

1.2 算法的描述和分析

1.2.1 算法的描述

算法需要用一种语言来描述,同时,算法可有各种描述方法以满足不同的需求。例如,一个需要在计算机上运行的程序(程序也是算法),必须严格按照语法规定用机器语言、汇编语言或高级语言编写;而一个为了供人阅读和交流的算法,可以用伪码语言或框图等其他形式来描述。伪码语言是一种包含高级程序语言的3种基本控制结构(顺序、分支和循环)和自然语言成分的“面向对象”的语言。

本书采用C语言描述数据结构及相应的算法。

1.2.2 算法的分析

1. 从时间方面分析

算法执行时间须通过依据该算法编制的程序在计算机上运行时所消耗的时间来度量。一个算法是由控制结构(顺序、分支和循环3种)和原操作(指固有数据类型的操作)构成的,因此算法时间取决于两者的综合效果。为了便于比较同一问题的不同算法,通常的做法是,从算法中选取一种对于所研究的问题(或算法类型)来说是基本操作的原操作,以该基本操作重复的次数作为算法的时间复杂度。算法的时间复杂度通常记为:

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

其中, n 为问题的规模 (或大小), $f(n)$ 表示算法中基本操作重复执行的次数, 是问题规模 n 的某个函数。上式表示随问题规模 n 的增大, 算法执行时间的增长率与 $f(n)$ 的增长率相同。 $f(n)$ 和 $T(n)$ 是同数量级的函数, 大写字母 O 表示 $f(n)$ 与 $T(n)$ 只相差一个常数倍。

算法的时间复杂度用数量级的形式表示后, 一般可简化为分析循环体内基本操作的执行次数即可。

【例 1-1】

```
(1) x++;
(2) for(i=1; i<=n; i++) x++;
(3) for(i=1; i<=n; i++)
    for(j=1; j<=n; j++) x=x+1;
```

上面 3 个程序段含基本操作“x 增 1”的语句 $x++$ 的频度分别为 1、 n 和 n^2 , 则其时间复杂度分别为 $O(1)$ 、 $O(n)$ 和 $O(n^2)$, 它们分别称为常量阶、线性阶和平方阶。其他算法的时间复杂度还有对数阶 $O(\log_2 n)$ 、指数阶 $O(2^n)$ 等。

【例 1-2】

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

程序段中 $x++$ 的时间复杂度为 $O(n^3)$ 。

```
(2) for(i=1; i<n; i++)
    for(j=1; j<i; j++)
        for(k=1; k<j; k++)
            x++;
```

程序段中 $x++$ 的时间复杂度为 $O(\sum_{i=1}^n \frac{i(i+1)}{2})$ 。

2. 从空间方面分析

类似于算法的时间复杂度, 以空间复杂度作为算法所需存储空间的量度, 记为:

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

其中, n 为问题的规模 (或大小), $S(n)$ 表示算法在运行过程中所需辅助存储空间的大小。这里不多作讨论。

1.3 典型例题

1. 通常要求同一逻辑结构中的所有数据元素具有相同的特性, 这意味着 ()。

- A. 数据元素具有同一特点
- B. 不仅数据元素所包含的数据项的个数要相同, 而且对应数据项的类型也要一致
- C. 每个数据元素都一样
- D. 数据元素所包含的数据项的个数要相等

[分析] 同一逻辑结构中的所有数据元素必须具有相同的数据域, 而且对应数据项的类型要一致。一方面, 只有同一类型的数据元素才能归类到同一结构中; 另一方面, 计算机存储器在表示数据元素时, 必须为它们定义相同的数据类型。故 B 答案是正确的。

[答案]B

2. 下列程序段的时间复杂度是多少?

```

...
for(i=1;i<=n;i++)
{
    ...
    k++;
    for(j=1,j<=n;j++) k+=1;
    ...
}
...

```

[分析]确切地计算一个算法的时间消耗没有任何实际意义。因为在具体问题中,问题规模 n 的大小是变化的,而且每台计算机的主频也不相同,所以应关注的是算法中每条语句的执行次数(频度)。对上述程序段,语句执行的总次数为: $n+1+n+n(n+1)+n^2=2n^2+3n+1$, 令 $T(n)=2n^2+3n+1$ 。

$T(n)$ 能反映出程序段的时间消耗。当 n 较大时, $T(n)$ 的大小主要取决于 n 的数量级。借助数学上极限的概念,如果 $\lim_{n \rightarrow \infty} \frac{T(n)}{f(n)} = \text{常数}$,表示当 $n \rightarrow \infty$ 时, $T(n)$ 和 $f(n)$ 具有相同的数量级,称 $T(n)$ 与 $f(n)$ 同阶。

在本例中,假设 $f(n) = n^2$, 则有 $\lim_{n \rightarrow \infty} \frac{T(n)}{n^2} = \text{常数}$

即 $T(n)$ 与 n^2 同阶。采用 O 记法,则有 $T(n)=O(f(n))=O(n^2)$ 。

其中, $T(n)$ 就是算法的渐近时间复杂度,简称时间复杂度,而 $f(n)$ 为算法中执行频度最大的那条语句的频度。由此得到求解时间复杂度的方法:找出所有语句中执行频度最大的那条语句的频度,然后取其数量级放入 O 中即可。

[答案] $O(n^2)$

习 题 1

一、选择题

- 数据的基本单位是 ()。
 - 数据结构
 - 文件
 - 数据元素
 - 数据项
- 以下哪种数据结构中的任何两个结点之间都没有逻辑关系 ()。
 - 树形结构
 - 集合
 - 图状结构
 - 线性结构
- 数据结构被形象地定义为 (D,R) , 其中 D 是 () (1) 的有限集合, R 是 D 上 () (2) 的有限集合。
 - (1) A. 算法
 - B. 数据元素
 - C. 数据操作
 - D. 逻辑结构
 - (2) A. 操作
 - B. 映像
 - C. 存储
 - D. 关系
- 在数据结构中,从逻辑上可以把数据结构分为 ()。
 - A. 动态结构和静态结构
 - B. 紧凑结构和非紧凑结构
 - C. 线性结构和非线性结构
 - D. 内部结构和外部结构

5. 线性表的线性结构是一种 () 的存储结构。
 A. 随机存取 B. 顺序存取 C. 索引存取 D. HASH 存取
6. 计算机算法指的是 () (1), 它必须具备输入、输出、() (2) 5 个特征。
 (1) A. 计算方法 B. 排序方法
 C. 解决某一问题的有限运算序列 D. 调度方法
 (2) A. 可行性、可移植性和扩充性 B. 可行性、确定性和有穷性
 C. 确定性、有穷性和稳定性 D. 易读性、稳定性和安全性
7. 线性表若采用链表存储结构, 要求内存中可用存储单元的地址 ()。
 A. 必须是连续的 B. 部分必须是连续的
 C. 一定是不连续的 D. 连续不连续都可以
8. 以下说法正确的是 ()。
 A. 线性表的线性存储结构优于链式存储结构
 B. 二维表组是它的每个数据元素为一个线性表的线性表
 C. 栈的操作方式是先进先出
 D. 队列的操作方式是先进后出
9. 以下说法正确的是 ()。
 A. 数据元素是数据的最小单位
 B. 数据项是数据的基本单位
 C. 数据结构是带有结构的各数据项的集合
 D. 数据结构是带有结构的数据元素的集合

二、填空题

1. 所谓数据的逻辑结构, 指的是数据元素之间的_____。
2. 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, 它包含_____、_____和_____ 3 方面内容。
3. 数据的逻辑结构包括_____、_____、_____和_____ 4 种类型。
4. 算法的 5 个重要特性是_____、_____、_____、_____、_____。
5. 下列程序段的时间复杂度是_____。
- ```
for(i=1;i<=n;i++)
 A[i]=0;
```
6. 下列程序段的时间复杂度是\_\_\_\_\_。
- ```
s=0;
for(i=1;i<=n;i++)
  for(j=1;j<=n;j++)
    s=s+b[i][j];
sum=s;
```
7. 存储结构是逻辑结构的_____实现。
8. 常见时间复杂度的量级有: 常数阶 O _____, 对数阶 O _____, 线性阶 O _____, 平方阶 O _____和指数阶 O _____。通常认为, 具有指数阶量级的算法是_____的。
9. 数据对象是性质相同的_____集合。

三、应用题

1. 分析下列程序段的时间复杂度。

```
...
i=1;
While(i<=n)
    i=i*2;
...
```

2. 叙述算法的定义及其重要特性。
3. 解释下列术语: 数据, 数据元素, 数据结构, 数据对象。
4. 逻辑结构与存储结构是什么关系?
5. 设 n 为正数, 试确定下列程序段中, 前面加记号@的语句的频度。

```
i=1;k=0;
While(i<=n-1)
{ @k+=10*i;
  i++;}
k=0;
for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
        @k++;
```

四、算法设计题

1. 已知输入 x 、 y 、 z 这 3 个不相等的整数, 设计一个算法, 使得这 3 个数按从大到小的顺序输出, 并考虑所用算法的比较次数和元素移动次数。
2. 编写算法实现: 在输入确定的 10 个数中, 找出最小的数和最大的数。

第2章 线性表

线性表的特点是：在数据元素的非空有限集中，存在唯一的一个被称为“第一个”的数据元素；存在唯一的一个被称为“最后一个”的数据元素；除第一个以外，集合中的每一个数据元素均有且只有一个前驱；除最后一个以外，集合中每一个数据元素均有且只有一个后继。

2.1 线性表的逻辑结构

线性表是最简单的一种数据结构。简单地说，一个线性表是 n ($n \geq 0$) 个具有相同属性的数据元素的有限序列，其中各个数据元素有着依次相邻（串接）的逻辑关系。

线性表中数据元素的个数 n 称为线性表的长度。 $n=0$ 时，该线性表为空表。 $n>0$ 时，该线性表可记为：

$$(A_0, A_1, \dots, A_i, \dots, A_{n-1})$$

其中， A_0 称为起点， A_{n-1} 称为终点，每个元素的序号代表它在该线性表中的逻辑位置减 1（与数组下标对应）。除了起点 A_0 和终点 A_{n-1} 外，每个数据元素都有且只有一个直接前驱和一个直接后继。 A_{i-1} 是 A_i 的直接前驱， A_{i+1} 是 A_i 的直接后继。

线性表中的数据元素可以是各式各样的，但同一线性表中的元素必定有相同的特性，即属于同类数据对象，且相邻数据元素之间存在着有序关系。

线性表是一个相当灵活的数据结构，其表长可根据不同的操作增长或缩短。对线性表进行的基本操作有：存取、插入、删除、合并、分解、查找、排序、求线性表的长度等。

2.2 线性表的顺序存储结构

2.2.1 顺序分配

线性表的顺序存储结构用一组地址连续的存储单元依次存储该线性表中的各个元素。假设线性表的每个数据元素占用 L 个存储单元，并以所占的第一个单元的存储地址为数据元素的存储位置，则第 $i+1$ 个数据元素的存储位置为：

$$\text{Loc}(A_i) = \text{Loc}(A_0) + i \times L$$

因此，在内存中可以通过地址计算直接存取线性表中的任一元素，即线性表的顺序存储结构可以随机存取。这种结构的特点是，逻辑上相邻的元素物理上也相邻，如图 2-1 所示。顺序表可用 C 语言的一维数组实现，数组的类型随数据元素的属性而定。

存储地址	内存状态	数据元素在线性表中的位置
$Loc(A_0)$	A_0	1
$Loc(A_0)+L$	A_1	2
$\vdots$	$\vdots$	$\vdots$
$Loc(A_0)+i \times L$	A_i	$i+1$
$\vdots$	$\vdots$	$\vdots$
$Loc(A_0)+(n-1) \times L$	A_{n-1}	n
	$\vdots$	

图 2-1 线性表的顺序存储结构

2.2.2 线性表的操作

线性表结构简单、易于实现，但插入或删除一个元素时，需对其后继的全部元素逐个进行移动（平均需移动表中的一半元素），操作不方便，且需花费较多时间，特别是当插入元素需要动态扩大连续存储时，线性表后面的存储区可能已被占用，从而无法扩大。所以，这种结构仅适用于数据元素个数不经常变动或只需在顺序存取设备上做成批处理的情况。下面分情况讨论线性表的插入和删除操作。

1. 线性表的插入操作

该操作有以下两种：

（1）在数组中下标为 i 的元素前插入一个新元素。

【例 2-1】某 C 语言程序中，整型数组 v 的 99 个元素 $v[0] \sim v[98]$ 组成一个线性表。为了在 $v[i]$ 位置前插入一个新元素 b ，可用函数 `inst1` 来实现，当插入成功时返回 1，否则返回 0，所以该函数的返回值类型是整型。插入前后的线性表的变化如图 2-2 和图 2-3 所示。

数据元素	下标
67	0
55	1
13	2
46	$\vdots$
32	$\vdots$
45	i
78	$\vdots$
23	$j-1$
	j
	99

图 2-2 插入前的数组

数据元素	下标
67	0
55	1
13	2
46	$\vdots$
32	$\vdots$
b	i
45	$\vdots$
78	$\vdots$
23	99

图 2-3 插入后的数组

分析：① 初始条件： v, i, b 。

② 执行条件： $0 \leq i \leq 98$ 。

③ 执行结果：1，成功。

0，不成功。