



高等学校用书·教材

GAODENG XUEXIAO YONGSHU JIAOCAI

数据结构实践 指导教程 (C 语言版)

湖北教育学院教材建设委员会 组编
主编 阮宏一

华中科技大学出版社
<http://press.hust.edu.cn>

高等学校用书·教材

GAODENG XUEXIAO YONGSHU JIAOCAI

数据结构实践 指导教程 (C 语言版)

湖北教育学院教材建设委员会 组编

主 编 阮宏一

副主编 杜发启 史毓达 文中林

华中科技大学出版社
<http://press.hust.edu.cn>

图书在版编目(CIP)数据

数据结构实践指导教程/湖北教育学院教材建设委员会 组编 阮宏一 主编
武汉:华中科技大学出版社,2004年9月
ISBN 7-5609-3211-8

I. 数…
II. ①湖… ②阮…
III. 数据结构-高等学校-教材
IV. TP311.2

湖北教育学院教材建设委员会 组编
数据结构实践指导教程(C语言版) 阮宏一 主编

责任编辑:姚 幸

封面设计:潘 群

责任校对:封春英

责任监印:张正林

出版发行:华中科技大学出版社

武昌喻家山 邮编:430074 电话:(027)87557437

录 排:华中科技大学惠友文印中心

印 刷:华中科技大学印刷厂

开本:787×1092 1/16

印张:23.5

字数:490 000

版次:2004年9月第1版

印次:2004年9月第1次印刷

定价:35.00元

ISBN 7-5609-3211-8/TP·531

湖北教育学院教材建设委员会

主 任 严文清
副主任 余吴恩 鲁晓成
委 员 (以姓氏笔画为序)

王志民	刘鸿翔	杜卓选	肖 明
李 明	陈洪波	宋移安	周立人
周和义	邵则遂	杨 云	张祖春
袁先溥	侯汉娜	梁治洲	曹 曼
黄为秀	熊华生		

内 容 提 要

本书是“数据结构”课程的辅助教材。全书分为上、下两篇。上篇“数据结构课程实践”，配合清华大学出版社出版的由严蔚敏、吴伟民编著的《数据结构（C语言版）》教材的内容进行编写，共由8章组成。每章分基本知识点、内容要点及课程实践3部分，其中实践部分给出了教材中主要算法的实现程序。下篇“数据结构应用实践”由若干应用开发程序组成，目的是为开拓学生学习视野、提高学生分析和解决问题的能力及实际应用开发的能力起到一个促进的作用。

本书是在作者多年教授“数据结构”课程，指导学生上机实践所积累的知识与经验的基础上编写而成的。所有程序都给出了源代码，在 Borland C++ Version 3.1 for DOS 和 Microsoft Visual C++ 6.0 下运行通过。

本书适用于计算机专业的学生及其他相关专业的学生学习“数据结构”课程上机实践使用，对软件开发爱好者亦是一本很好的参考书。

总 序

教材是教学依据的蓝本，是教育的基本思想和培养目标的具体体现。我们首次组织编写这套系列教材和教辅资料，力求完美体现湖北教育学院的办学思想、育人原则和治学精神，以形成特色，铸造精品。

以学生发展为本进行人性化设计，实现素质教育是教材编写工作的目标追求。教材的编写应遵循学生的身心发展规律，利于引导他们利用已有的知识与经验，主动探索新的知识，培养他们的创新精神和实践能力，促进他们良好个性的发展。教材的编写应充分体现学生观，变学生从教材的被动受体为对教材进行能动实践的创造主体；提供丰富的与学生生活背景有关的资料，从学生已有的经验和兴趣出发，让学生通过亲身体验探索、思考和研究的过程，激发学生的学习兴趣；积极引导学生把所学知识应用于实际，从本学科角度对某些日常生活、生产和其他学科中出现的问题进行探究。教材不仅只对教育经验进行完美预设，也应该给学生留有发展、创造的空间；不仅传授知识，更要教给学生思考、分析、处理问题的方法，使学生在课余时间也可以自主地进行学习和深化，成为素质教育的“学材”。

最大程度地实现教师的专业创造和教学创新是教材编写的价值追求。教材既是对教师多年教学实践的总结，又是在总结基础上的再创造。因此，在教材的组织编排设计中，要根据社会实践和学科发展的状况，在体系上有所创新，在内容上有所更新，在教法和学法上有所革新。把教师多年来在教学形成的教学策略和教学方法融合进去，把教师的专业创造、教学创新体现出来，使之成为教师教学科研的平台。

教材还是教师对教学进行构建和创造的主要媒体，在教材的编写手段上，应当针对具体的教学内容，在把握好教学策略和方法的同时，改变那种将所有事实和原理全部直接呈现的方式，便于教师运用启发式教学方式，完善教学过程。

逐步发展多样化的教材形式，是教材编写工作的新领域和努力方向。随着科学技术的发展，教学手段不断进步，教材的概念范围有了新的延伸，正在从静态向动态快速发展，信息技术以其巨大的影响力正在改变着人们获取知识的途径与方式。单纯的文字教

材已不能承载全部的教学任务, 满足教学和学生的需求。因此, 还应当根据本学科的特点和要求, 根据学生的身心发展水平设计、开发、制作出学生学习和技能训练必不可少的网络信息和电子教材, 以及与之相配套的练习册、操作材料、图表、实验报告和其他读物, 为教学精心打造和提供可利用的课程资源。这是教材编写工作面临的挑战和艰巨任务。

此次教材组织开发是湖北教育学院教材建设的良好开端, 要把教材开发作为湖北教育学院的一项战略性任务继续深入下去, 期盼更多的教师加入这个行列, 为教学质量的提高做出新贡献。

本套教材在编写出版过程中, 得到有关单位、专家、学者的大力支持和帮助, 在此谨表谢意。



2004 年 8 月

前言

“数据结构”是计算机软件的一门重要的基础课程。计算机科学各领域及有关的应用软件都要用到各种数据结构。语言编译要使用栈、散列表及语法树；操作系统中使用队列、存储管理表及目录树；数据库系统要运用线性表、多链表及索引树等进行数据管理；而在人工智能领域根据求解问题性质的差异将涉及各种不同的数据结构，如广义表、集合、搜索树及各种有向图等。

“数据结构”课程的目的是教学生学会分析研究各种数据结构及其内在的逻辑关系，掌握它们在计算机中的存储表示及其相应的算法，并初步掌握算法的时间分析和空间分析的技术。由于它对计算机学科起到承前启后的作用，因此学生学好数据结构这门课程至关重要。

数据结构的原理及算法较抽象，而该课程又一般在本科低年级开设，我们在多年的教学过程中深深感到，对于只具有计算机程序设计基础知识的学生，要他们很好地理解和掌握其中的原理比较困难，特别是理解数据的存储结构、相关算法，以及将算法转化成相应的应用程序，这些往往是学习的难点。针对这种情况，我们编写了这本《数据结构实践指导教程（C语言版）》，供计算机专业的学生及其他相关专业的学生上机参考。希望能够帮助学生在尽可能短的时间内，对数据结构的知识与应用有一个比较全面、深入和系统的认识，为进一步提高学生的应用编程能力在方法上提供指导。

本书分为上、下两篇。

上篇为数据结构课程实践，共 8 章。配合清华大学出版社出版的由严蔚敏、吴伟民编著的《数据结构（C语言版）》教材的内容进行编写。针对该教材各主要章节，每章分为 3 部分：基本知识点部分，给出该章应该掌握的基本术语；内容要点部分，从实践的角度总结相应数据结构的基本概念、基本运算、存储结构及相应的图示；实践部分，给出教材中主要算法的全部实验，也是该书的重点内容，对每个实验，均给出了实验目的、实验内容、实验要点及说明、参考程序，在每个参考程序之后给出了一两个思考题，希望学生能在实践过程中，自己动手完成，达到提高编程能力的目的。

下篇为数据结构应用实践，即第 9 章。给出了 6 个数据结构的应用开发程序，实现了教材中各种数据结构在实际应用开发中的应用。每个应用程序分问题描述、实现功能、

算法思想、数据结构、参考程序及测试结果 6 个部分, 希望能开拓学生学习视野、提高学生分析问题和解决问题的能力。

为了更好地配合严蔚敏、吴伟民编著的《数据结构 (C 语言版)》教材, 本书的源程序采用了两种编写形式。部分源程序完全用标准 C 语言实现, 另一部分源程序在 C 语言描述的基础之上, 采用了 C++ 语言的引用调用参数传递方式, 使程序结构更为简明清晰。读者只需对 C++ 的引用调用参数的传递方式作一个了解, 就可以读懂书中的所有源程序。本书的所有程序都在 Borland C++ Version 3.1 for DOS 和 Microsoft Visual C++ 6.0 下运行通过。方便学生在不同的环境使用。

本书第 1 章、第 2 章、第 4 章、第 6 章由阮宏一、夏汇川编写; 第 3 章、第 8 章由杜发启编写; 第 5 章、第 7 章由文中林编写; 第 9 章中, 中序表达式求值、迷宫问题由文中林编写, Josephu 环、哈夫曼编码、简单文本编辑器实现、年级学生成绩管理系统由史毓达、张晓蓉编写。

本书的出版, 得到了湖北教育学院计算机系、湖北教育学院教务处、湖北教育学院期刊社的大力支持, 在此表示衷心的感谢!

由于作者水平有限, 书中难免存在错误和疏漏之处, 敬请广大读者批评指正。

作 者
2004 年 6 月

目 录

上篇：数据结构课程实践

第 1 章 线性表	(3)
1.1 内容要点	(3)
1.1.1 线性表的定义及基本运算	(3)
1.1.2 线性表的存储结构	(4)
1.2 线性表的实践	(7)
1.2.1 顺序表的实践	(7)
实践一 顺序表的建立	(7)
实践二 顺序表的插入	(9)
实践三 顺序表的删除	(11)
实践四 顺序表的查找	(13)
1.2.2 链表的实践	(15)
实践一 单链表的建立	(15)
实践二 单链表的查找	(17)
实践三 单链表的插入	(19)
实践四 单链表的删除	(22)
实践五 单链表的合并	(24)
实践六 双向链表的建立及遍历	(27)
实践七 循环单链表的建立及查找	(30)
实践八 双向循环链表的建立及插入	(32)
实践九 双向循环链表的查找及删除	(36)
第 2 章 栈和队列	(39)
2.1 内容要点	(39)
2.1.1 栈	(39)
2.1.2 队列	(41)
2.2 栈的实践	(44)

实践一 顺序栈的建立.....	(44)
实践二 顺序栈的插入.....	(46)
实践三 顺序栈的共用.....	(48)
实践四 链式栈的建立.....	(52)
2.3 队列的实践.....	(55)
实践一 顺序循环队列的实现.....	(55)
实践二 链队列的实现.....	(58)
第3章 串	(62)
3.1 内容要点.....	(62)
3.1.1 串的定义及基本运算.....	(62)
3.1.2 串的存储结构.....	(63)
3.1.3 串的模式匹配算法.....	(64)
3.2 串实践.....	(67)
实践一 用定长顺序存储表示的串连接操作.....	(67)
实践二 求子串.....	(70)
实践三 堆分配存储表示的串操作.....	(72)
实践四 串的简单模式匹配.....	(79)
实践五 串的改进模式匹配.....	(81)
第4章 多维数组、矩阵和广义表	(84)
4.1 内容要点.....	(84)
4.1.1 多维数组.....	(84)
4.1.2 特殊矩阵、稀疏矩阵.....	(86)
4.1.3 广义表.....	(89)
4.2 多维数组、矩阵和广义表实践.....	(92)
实践一 数组的顺序表示和实现.....	(92)
实践二 稀疏矩阵的转置.....	(97)
实践三 稀疏矩阵的求和、求积运算.....	(103)
实践四 广义表的递归算法.....	(115)
第5章 树与二叉树	(123)
5.1 内容要点.....	(123)
5.1.1 树和森林.....	(123)
5.1.2 二叉树.....	(125)
5.2 二叉树的实践.....	(130)
实践一 二叉树的建立.....	(130)
实践二 二叉树的前序遍历.....	(133)
实践三 二叉树的中序遍历.....	(134)

实践四	二叉树的后序遍历.....	(135)
实践五	二叉树的层次遍历.....	(136)
实践六	求二叉树的叶子个数.....	(138)
实践七	二叉树的复制.....	(140)
实践八	二叉排序树的结点删除.....	(142)
实践九	线索二叉树的建立及遍历.....	(148)
实践十	哈夫曼树的建立.....	(152)
第6章	图.....	(156)
6.1	内容要点.....	(156)
6.1.1	图的定义及基本运算.....	(156)
6.1.2	图的存储结构.....	(157)
6.1.3	图的遍历.....	(161)
6.1.4	图的应用.....	(161)
6.2	图的实践.....	(164)
6.2.1	图的建立及遍历.....	(164)
实践一	无向图的邻接矩阵表示及遍历.....	(164)
实践二	有向图的邻接表表示及遍历.....	(171)
实践三	有向图的十字链表表示及遍历.....	(178)
6.2.2	图的应用实践.....	(186)
实践一	构造无向图的生成树.....	(186)
实践二	构造无向连通网的最小生成树.....	(191)
实践三	有向图的拓扑排序.....	(195)
实践四	找最短路径.....	(200)
第7章	查找.....	(205)
7.1	内容要点.....	(205)
7.1.1	顺序查找.....	(205)
7.1.2	折半查找.....	(206)
7.1.3	分块查找.....	(206)
7.1.4	二叉排序树查找.....	(206)
7.1.5	平衡二叉树查找.....	(206)
7.1.6	哈希 (Hash) 表查找.....	(207)
7.2	查找实践.....	(207)
实践一	顺序表的顺序查找.....	(207)
实践二	链表的顺序查找.....	(208)
实践三	折半查找.....	(210)
实践四	分块查找.....	(213)
实践五	二叉排序树查找.....	(215)

实践六 哈希查找(1)	(218)
实践七 哈希查找(2)	(221)
第8章 排序	(225)
8.1 内容要点	(225)
8.1.1 排序的基本概念	(225)
8.1.2 记录序列的存储方式	(226)
8.2 内部排序	(226)
8.2.1 插入排序	(226)
8.2.2 交换排序	(227)
8.2.3 选择排序	(228)
8.2.4 归并排序	(229)
8.2.5 基数排序	(229)
8.3 排序实践	(230)
实践一 直接插入排序	(230)
实践二 折半插入排序	(233)
实践三 表插入排序	(234)
实践四 希尔排序	(239)
实践五 冒泡排序	(241)
实践六 快速排序	(242)
实践七 简单选择排序	(245)
实践八 堆排序	(246)
实践九 归并排序	(248)
实践十 链式基数排序	(251)

下篇：数据结构应用实践

第9章 数据结构应用实例	(261)
9.1 中序表达式的计算	(261)
9.2 迷宫问题	(267)
9.3 Josephu 环	(275)
9.4 哈夫曼编码	(282)
9.5 简单文本编辑器实现	(290)
9.6 年级学生成绩管理系统	(320)
参考文献	(363)



数据结构课程实践

第 1 章

线 性 表

基本知识点

线性表

顺序表的操作

双链表

头结点

线性表的顺序存储结构

线性表的链式存储结构

循环链表

单（双）链表的基本操作

顺序表

单链表

头指针

静态链表

1.1 内 容 要 点

1.1.1 线性表的定义及基本运算

线性表是最基本最常用的数据结构。简单地说，一个线性表是 n 个**同类型**数据元素的有限序列并且相邻数据元素之间存在着**序偶**关系。通常记为：

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

其中 n 为表长， $n=0$ 时称为空表。表中相邻元素之间存在着顺序关系。将 a_{i-1} 称为 a_i 的**直接前驱**， a_{i+1} 称为 a_i 的**直接后继**。

线性表是一种线性结构。线性结构的特点是数据元素之间满足一种线性关系，数据元素“一个接一个地排列”。线性结构在数据元素的非空集合中有以下特性：

- 存在唯一一个称为“第一个”的元素；
- 存在唯一一个称为“最后一个”的元素；
- 除第一个元素之外，集合中的每个元素都只有一个直接前驱；
- 除最后一个元素之外，集合中的每个元素都只有一个直接后继。

在线性表上的基本操作有：

- (1) InitList (&L) —— 构造一个空的线性表L；
- (2) ListLength (L) —— 返回线性表 L 中所含数据元素的个数；
- (3) GetElem(L, i, &e) —— 用 e 返回线性表 L 中第 i 个数据元素的值；
- (4) LocateElem (L, e, compare()) —— 在表 L 中查找与值 e 满足 compare() 关系的

第一个数据元素的序号, 若在 L 中未找到这样的数据元素, 返回 0 值;

(5) $ListInsert(&L, i, e)$ —— 在线性表 L 的第 i 个元素之前插入一个值为 e 的新元素, 使原序号为 $i, i+1, \dots, n$ 的数据元素的序号变为 $i+1, i+2, \dots, n+1$, 插入后 L 的长度加 1;

(6) $ListDelete(&L, i, &e)$ —— 在线性表 L 中删除第 i 个数据元素, 并用 e 返回其值。删除后使序号为 $i+1, i+2, \dots, n$ 的元素序号变为 $i, i+1, \dots, n-1$, 删除后 L 的长度减 1。

1.1.2 线性表的存储结构

1. 线性表的顺序存储

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储线性表中的各个数据元素。在这种顺序存储结构中, 线性表中**逻辑**上相邻的两个元素在**物理位置**上也相邻。

一般来说, 线性表的第 i 个数据元素 a_i 的存储位置为

$$LOC(a_i) = LOC(a_1) + (i-1) * L$$

其中 $LOC(a_1)$ 是线性表的**第一个**数据元素 a_1 的存储位置, 通常称为线性表的**起始位置**。 L 为表中每个元素所占存储空间的大小。

线性表的顺序存储结构是一种**随机存储**结构, 容易实现随机存取线性表的第 i 个数据元素的操作, 但在实现插入、删除等操作时需要移动大量的数据元素。所以, 它适用于数据相对稳定的数据表, 例如职工工资表、学生学籍表等。

线性表的顺序存储结构示意如图 1-1 所示。以这种顺序存储结构表示的线性表常称为**顺序表**。

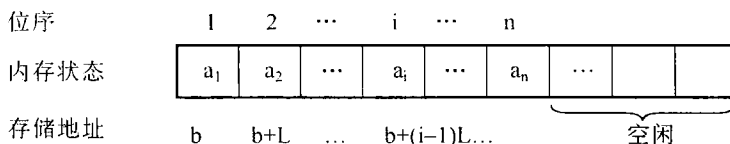


图 1-1 线性表的顺序存储结构

实际应用中, 常采用动态分配的一维数组来表示顺序表, 其存储结构定义如下。动态分配的结构示意如图 1-2 所示。

```
typedef struct          // 顺序表结构
{
    ElemType *elem;      // 存储空间的基址
    int length;          // 当前表长
    int listsize;        // 当前已分配的存储空间 (按元素大小为单位)
} SqList;
```

2. 线性表的链式存储

线性表的链式存储是用一组任意的存储单元存储线性表的数据元素。这组存储单元可以是连续的, 也可以是不连续的。因此, 元素之间的逻辑关系需要用指针来表示。这种用链接的方法存储的线性表称为**链表**。