

国 家 级 教 学 成 果 奖 配 套 教 材

数据结构学习与实验指导

Learning and Experimental Guidance of Data Structures

陈 越 何钦铭 徐镜春 魏宝刚 杨 彬 编著



高等教育出版社
HIGHER EDUCATION PRESS

国家级教学成果奖配套教材

数据结构学习与实验指导

Shuju Jiegou Xuexi yu Shiyan Zhidao

陈 越 何钦铭 徐镜春 魏宝刚 杨 彬 编著



高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容提要

本书是《数据结构》(陈越主编,高等教育出版社2012年出版)的配套教材,围绕教材中的主要知识点设计了27道“实验案例”和38道“实验项目”,并提供详细解答。

第1章概要介绍了本书的特点和内容结构;第2章复习C语言的关键内容;第3章针对线性表的知识点设计应用问题;第4章围绕树的存储、重要性质与应用进行练习;第5章是针对散列表和经典哈希映射技术的应用;第6章涉及图的各种表示方法和相关算法;第7章通过对各种类型的大规模排序问题的求解,帮助读者理解各种经典排序算法的特点和适用范围;第8章给出的题目涉及多个知识点的综合应用,帮助读者深刻体会数据结构灵活运用。

本书可作为“数据结构”课程的参考用书。读者可在精读“实验案例”的基础上,自己动手实现一部分“实验项目”,锻炼分析与解决问题的能力;还可以尝试解决每个实验后面给出的略带研究性质的思考题,提高独立思考和研究的能力。

图书在版编目(CIP)数据

数据结构学习与实验指导 / 陈越等编著. --北京:高等教育出版社,2013.5

ISBN 978 - 7 - 04 - 037217 - 5

I. ①数… II. ①陈… III. ①数据结构 - 高等学校 - 教学参考资料②C语言 - 程序设计 - 高等学校 - 教学参考资料 IV. ①TP311.12②TP312

中国版本图书馆CIP数据核字(2013)第072896号

策划编辑 张 龙
插图绘制 尹文军

责任编辑 张 龙
责任校对 刁丽丽

封面设计 杨立新
责任印制 尤 静

版式设计 马敬茹

出版发行 高等教育出版社
社 址 北京市西城区德外大街4号
邮政编码 100120
印 刷 北京市密东印刷有限公司
开 本 787mm × 1092mm 1/16
印 张 16.25
字 数 360千字
购书热线 010 - 58581118

咨询电话 400 - 810 - 0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landraco.com>
<http://www.landraco.com.cn>
版 次 2013年5月第1版
印 次 2013年5月第1次印刷
定 价 28.00元(含光盘)

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换

版权所有 侵权必究
物 料 号 37217 - 00

前言

“数据结构”是计算机类专业最重要的专业基础课之一,主要教授数据的有效组织方法以及解决实际问题的各种经典算法。在“数据结构”课程学习中存在这样一个问题:虽然了解了重要的数据结构和配套的算法,但在遇到实际问题时,对于什么是最佳的数据结构和算法往往难以判断和选择。一般的验证性的课后习题,由于问题过于简单,或只是纸上谈兵,没有提供大数据量测试的条件,很难培养对算法时间、空间复杂度的感性认识。而经典算法的威力,往往是在处理大规模数据量时才能真正体现。只有自己动手解决规模较大的问题,才能建立起感性认识,更好地理解数据结构和算法存在的意义。

本书是《数据结构》(陈越主编,高等教育出版社 2012 年出版)的配套教材。主教材的特点是从实际应用问题出发,导出各种经典数据结构的定义、实现(存储)方法以及操作实现,最后以丰富的综合应用案例帮助读者理解这些数据结构为什么存在,以及在什么情况下可以最好地解决什么样的问题。本书围绕主教材中的主要知识点设计了 27 道“实验案例”,每一个实验案例都给出详细的测试数据、解答和源代码,帮助读者从实际应用的角度更好地理解知识点。另外还设计了 38 道“实验项目”,作为“实验案例”的延伸和提高,可以在实验案例已经提供的解答程序的基础上完成。读者可在精读“实验案例”的基础上,自己动手实现一部分“实验项目”,锻炼分析与解决问题的能力;还可以尝试解决每个实验后面给出的略带研究性质的思考题,提高独立思考和研究的能力。

第 1 章概要介绍了本书的特点和内容结构;第 2 章复习 C 语言的关键内容;第 3 章针对线性表的知识点设计应用问题;第 4 章围绕树的存储、重要性质与应用进行练习;第 5 章是针对散列表和经典哈希映射技术的应用;第 6 章涉及图的各种表示方法和相关算法;第 7 章通过对各种类型的大规模排序问题的求解,帮助读者理解各种经典排序算法的特点和适用范围;第 8 章给出的题目涉及多个知识的综合应用,帮助读者深刻体会数据结构的灵活运用。

本书提供了以下丰富的学习资源。

本书所附光盘中除了给出与主教材配套的电子课件外,还包括一个编程能力测试(Programming Ability Test, PAT)单机版的安装程序。PAT 单机版包含本书全部 27 道“实验案例”的题目及测试数据、标准解答程序。

为采用本书作为教材的任课教师提供可以安装在校园网内的“数据结构上机练习测试系统”。该测试系统附赠题库,允许管理员(Admin)自己增删题目,并且提供选择题题库管理、考试等功能,为任课教师统一组织学生进行上机练习和考试提供了方便的工具。任课教师可以通过邮箱 jsj@pub.hep.cn 与高等教育出版社联系,免费获取。

同时还提供 PAT 在线系统(网址 <http://pat.zju.edu.cn/>),全天候为广大读者提供免费服务,提供 35 道“实验项目”(不包括第 7 章的 3 道“开放性实验项目”)的在线评判功能;同时可以接受任课教师的申请,提供在线练习、考试等功能,并提供相应的统计功能。

本书由浙江大学计算机科学与技术学院教师编写:陈越教授撰写第 1、5、8 章,并负责全书统稿;何钦铭教授撰写第 2、3 章;魏宝刚教授撰写第 4 章;徐镜春副教授撰写第 6 章并制作配套电子课件;杨崧副教授撰写第 7 章。

特别感谢浙江大学计算机科学与技术学院研究生吴峻同学,他独立开发的 PAT 系统单机版和在线系统为本书提供了强大的技术支持,他的出色工作使得本书为读者提供更好的服务成为可能。

由于作者水平所限,书中不当之处在所难免,敬请广大读者批评指正。

编著者
2013 年 3 月

目 录

第 1 章 概论	1	与加法运算	54
第 2 章 数据结构实现基础	2	实验项目 3-5:求链式线性表的 倒数第 K 项	56
实验案例 2-1:简单计算器	4	实验项目 3-6:表达式转换	58
实验案例 2-2:数组元素循环右移 问题	7	实验项目 3-7:求前缀表达式 的值	60
实验案例 2-3:整数分解为若干项 之和	12	实验项目 3-8:堆栈模拟队列	62
实验案例 2-4:递增链表的插入	17	实验项目 3-9:队列中的 元素排序	64
实验项目 2-5:求集合数据的 均方差	20	第 4 章 树	66
实验项目 2-6:数列求和	22	实验案例 4-1:还原二叉树	68
实验项目 2-7:素因子分解	24	实验案例 4-2:树种统计	72
实验项目 2-8:用扑克牌计算 24 点	26	实验案例 4-3:朋友圈	76
实验项目 2-9:装箱问题模拟	29	实验案例 4-4:Windows 消息 队列	80
实验项目 2-10:海盗分赃	31	实验项目 4-5:家谱处理	85
实验项目 2-11:两个有序链表序列 的合并	34	实验项目 4-6:搜索树判断	89
实验项目 2-12:两个有序链表序列 的交集	36	实验项目 4-7:修理牧场	92
实验项目 2-13:两个有序序列的 中位数	38	实验项目 4-8:目录树	94
第 3 章 线性结构	41	实验项目 4-9:笛卡儿树	96
实验案例 3-1:一元多项式求导	42	第 5 章 散列查找	99
实验案例 3-2:汉诺塔的非递归 实现	46	实验案例 5-1:整型关键字的散列 映射	100
实验案例 3-3:银行业务队列简单 模拟	50	实验案例 5-2:字符串关键字的 散列映射	104
实验项目 3-4:一元多项式的乘法		实验案例 5-3:电话聊天狂人	109
		实验案例 5-4:基于词频的文件 相似度	115
		实验项目 5-5:QQ 账户的申请	

与登录	124	实验案例 7-3:统计工龄	197
实验项目 5-6:航空公司 VIP		实验案例 7-4:恢复古诗	200
客户查询	127	实验项目 7-5:魔法优惠券	206
实验项目 5-7:新浪微博		实验项目 7-6:奥运排行榜	208
热门话题	130	实验项目 7-7:PAT 排名汇总	211
实验项目 5-8:迷你搜索引擎	133	开放性题目 7-8:电话号码的磁盘	
第 6 章 图	136	文件排序	214
实验案例 6-1:哥尼斯堡的		开放性题目 7-9:Google 24 小时内	
“七桥问题”	138	的搜索关键字排行榜	215
实验案例 6-2:地下迷宫探索	143	开放性题目 7-10:BBS 帖子	
实验案例 6-3:六度空间	151	排序	216
实验案例 6-4:公路村村通	159	第 8 章 综合应用	217
实验案例 6-5:旅游规划	165	实验案例 8-1:打印选课	
实验项目 6-6:任务调度的		学生名单	218
合理性	169	实验案例 8-2:银行排队问题之单	
实验项目 6-7:哈利·波特的考试 ..	172	窗口“夹塞”版	225
实验项目 6-8:城市间紧急救援	175	实验案例 8-3:逆散列问题	236
实验项目 6-9:社交网络图中结点		实验项目 8-4:打印学生选课	
的“重要性”计算	178	清单	243
实验项目 6-10:关键活动	182	实验项目 8-5:银行排队问题之“多	
第 7 章 排序	186	队列多窗口”版	246
实验案例 7-1:模拟 Excel 排序	188	实验项目 8-6:畅通工程之局部最小	
实验案例 7-2:寻找大富翁	193	花费问题	250

第 1 章

概 论

“数据结构”是计算机类专业的重要专业基础课,一般介绍有关数据组织、算法设计、时间和空间效率的概念和通用分析方法。在解决实际问题时,选择合适的数据结构可以使算法的运行或存储效率更高;反之,选择了特定的算法后也需要设计合适的数据结构与之配合,达到最佳效果。所以在进行程序设计时,必须将数据结构和与之相关的算法结合起来考虑。通过该课程的学习,读者应学会数据的组织方法和现实世界问题在计算机内部的表示方法,针对问题的应用背景分析、选择合适的数据结构,从而培养高级程序设计技能。

要真正扎实掌握数据结构的知识,并且更重要的,要能根据实际情况灵活应用数据结构知识去解决问题,仅仅了解理论知识是远远不够的,需要大量的训练,才能在解决真实问题时得心应手。

《数据结构学习与实验指导》一书是“数据结构”课程的辅助教材,主要围绕数据结构的基本知识点,设计大量练习题目辅助读者学习。通过详细分析 27 道“实验案例”的解法,帮助读者完成从对知识点的理解,到应用其解决比较复杂的问题,再到动手实现解决问题的算法并设计实施有效的测试。同时还设计了 38 道“实验项目”(其中包括 3 道开放性题目)供读者自己思考并练习,从而达到全面提高分析问题、解决问题能力的目的。

后续每一章都会给出该章实验题目及涉及知识的列表,随后的内容基本上分两大部分。

(1) “实验案例”及详细分析与解答。包括对实验内容和输入输出的明确要求以及一套测试样例;对问题的分析以及编程实现的要点;给出完整的解题源代码;最后在此题基础上给出实验思考题供读者做进一步深入的思考。

(2) “实验项目”及分析与提示。同样包括对实验内容和输入输出的明确要求以及一套测试样例;与“实验案例”不同的是,这里只比较详细地给出解题思路、对问题的分析以及编程实现的要点,而把具体的编程实现留给读者自己去完成;最后也是以若干实验思考题结尾。

本书配套的光盘提供一个编程能力测试(PAT)的单机版,内含全部“实验案例”的测试用例,读者可在本地安装系统,并尝试提交自己的程序去解决问题。PAT 单机版中还提供了书中程序源代码作为参考答案,读者可以直接提交参考答案,观察其时间、空间效率,并尝试自己做局部优化,以得到更高效的解决方案。

全部“实验项目”可在浙江大学提供的 PAT 在线系统(网址 <http://pat.zju.edu.cn/>)找到。该系统全天候为广大读者提供免费服务,读者可自行注册账号,随时提交测试自己的解决方案。

第 7 章的开放性题目有多种解决方案,且有可能涉及海量数据,不适合使用自动判题系统评判。有兴趣的读者可以自行设计测试方案。

第 2 章

数据结构实现基础

本章实验内容主要围绕循环、数组、递归函数和链表这 4 部分内容,共包括了 4 个实验案例和 9 个实验项目。这些题目涉及的知识内容见表 2.1。其中,题目 2-3、2-8、2-10 比较难,题目 2-4、2-5、2-11、2-12 比较容易。

建议在学习中,选择 2 个实验案例进行深入学习与分析,再选择 2 个实验项目进行具体的编程实践。

表 2.1 第 2 章实验涉及的知识点

序号	题目名称	类别	内容	涉及主要知识点
2-1	简单计算器	实验案例	模拟只进行无优先级的整数加减乘除运算的简单计算器的工作	循环、输入处理、switch 控制语句
2-2	数组元素循环右移问题	实验案例	将数组中的 N 个整数循环向右移 M 个位置	循环、数组、宏定义
2-3	整数分解为若干项之和	实验案例	求正整数 N 的所有整数分解式子	循环、数组、递归函数
2-4	递增链表的插入	实验案例	构造递增序列对应的链表,插入给定 M 后使之仍保持有序	循环、链表
2-5	求集合数据的均方差	实验项目	求 N 个给定整数的均方差	循环、数组、浮点数格式化输出
2-6	数列求和	实验项目	求形如 $\{A, AA, AAA, \dots, AA \dots A\}$ 的数列之和	循环、数组、大整数的存储与运算
2-7	素因子分解	实验项目	输出给定正整数的素因式分解表达式	递归函数、递归的非递归实现
2-8	用扑克牌计算 24 点	实验项目	任取 4 张牌,添加运算符使计算结果为 24	嵌套循环、二维数组

续表

序号	题目名称	类别	内容	涉及主要知识点
2-9	装箱问题模拟	实验项目	将 N 个物品顺序装箱, 每个装入第一个能放入的箱子	嵌套循环、数组、贪心算法
2-10	海盗分赃	实验项目	P 个海盗分 D 颗钻石, 求第一个海盗所得	嵌套循环、二维数组、递归函数
2-11	两个有序链表序列的合并	实验项目	将两个有序链表合并为一个有序链表	循环、链表
2-12	两个有序链表序列的交集	实验项目	求两个非降序链表序列的交集序列	循环、链表
2-13	两个有序序列的中位数	实验项目	求两个等长的非降序序列的中位数	循环、二分查找

实验案例 2-1: 简单计算器

1. 实验目的

- (1) 熟练掌握用 while 语句控制循环。
- (2) 熟练理解和掌握字符型及整型数据的输入与处理。
- (3) 熟练掌握用 switch 语句控制多分支选择。

2. 实验内容

模拟简单计算器的工作。假设计算器只能进行加减乘除运算,运算数和结果都是整数,4 种运算符的优先级相同,按从左到右的顺序计算。

3. 实验要求

- (1) 输入说明:输入一个四则运算算式(没有空格,且至少有一个操作数),遇等号“=”说明输入结束。
- (2) 输出说明:输出算式的运算结果或错误信息。
- (3) 测试用例:

序号	输入	输出	说明
1	$1 + 2 * 10 - 10 / 2 =$	10	正常测试 4 种运算。
2	15 =	15	只有一个数字。
3	$32 / 0 =$	除法分母为零!	非正常退出。
4	$3 \% 4 =$	非法输入!	非正常退出。

4. 实验分析

(1) 问题分析

4 种运算符都是双目运算符,因此输入的四则运算算式的规律是“操作数 运算符 操作数 运算符 …”。因假定 4 种运算符的优先级相同,按从左到右的顺序计算,因此可先读入一个整数作为左操作数,然后根据读入的运算符决定所做的运算;当再输入一个整数(右操作数)后可计算出当前值,并作为下一次运算的左操作数,如此循环,直到输入的运算符是“=”时结束。

(2) 实现要点

操作数(整型)可用 scanf() 函数输入,而由于没有空格,运算符可用 getchar() 或者 scanf() 函数输入。用 switch 语句来控制实现各种运算符不同的运算。

5. 实验参考代码

```
#include <stdio.h>

int main()
{
    int Sum,X;
    char Op;
    scanf("%d%c",&Sum,&Op);
    /* 输入第一个操作数以及操作符;由于输入无空格,故无需特别处理 */
    while(Op != '=') {
        scanf("%d",&X);    /* 输入下一个操作数 */
        switch(Op) {
            case '+':        /* 根据操作符进行相应运算 */
                Sum += X;
                break;
            case '-':
                Sum -= X;
                break;
            case '*':
                Sum *= X;
                break;
            case '/':
                if ( X == 0 ) {
                    printf("除法分母为零! \n");
                    return 0;
                }
                Sum /= X;
                break;
            default:
                printf("非法输入! \n");
                return 0;
        }
        scanf("%c",&Op);    /* 输入下一个操作符 */
    }
    printf("%d\n",Sum);    /* 打印输出 */
    return 0;
}
```

代码 2.1 简单计算器

6. 实验思考题

- (1) 如果允许输入中出现空格等空白符,该如何处理输入? 如果允许第一个运算数含有“+”、“-”号(正负号),程序该怎样进行修改?
- (2) 如果运算符具有优先级,即先乘除后加减,该如何改进程序? [提示:加入子函数专门处理某个操作数开始的连续乘除法,并返回其运算结果]
- (3) 如果允许使用括号,该如何改进程序? [提示:采用递归函数]

实验案例 2-2:数组元素循环右移问题

1. 实验目的

- (1) 熟练使用循环语句。
- (2) 熟练理解和掌握数组存储与数据处理算法的关系。
- (3) 熟悉宏定义及其使用。

2. 实验内容

一个数组 A 中存有 $N(N>0)$ 个整数,在不允许使用另外数组的前提下,将每个整数循环向右移 $M(M\geq 0)$ 个位置,即将 A 中的数据由 $(A_0 A_1 \cdots A_{N-1})$ 变换为 $(A_{N-M} \cdots A_{N-1} A_0 A_1 \cdots A_{N-M-1})$ (最后 M 个数循环移至最前面的 M 个位置)。如果需要考虑程序移动数据的次数尽量少,要如何设计移动的方法?

3. 实验要求

- (1) 输入说明:第 1 行输入 $N(1\leq N\leq 100)$ 、 $M(M\geq 0)$;第 2 行输入 N 个整数。
- (2) 输出说明:输出循环右移 M 位以后的整数序列。
- (3) 测试用例:

序号	输入	输出	说明
1	6 2 1 2 3 4 5 6	5 6 1 2 3 4	一般情况
2	6 8 1 2 3 4 5 6	5 6 1 2 3 4	$M > N$ 的情况
3	3 6 11 23 56	11 23 56	$M > N$ 且正好是 N 的倍数
4	1 0 8	8	边界测试,最小的 N 和 M
5	令 N 为 100, M 为 99 输入数据略	略	边界测试,最大的 N 和 M

4. 实验分析

(1) 问题分析

输入的 N 个整数可以放在一个一维数组中。最容易想到的,也是简单的思路是循环右移一位的操作重复进行 M 次即可,但这种做法的数据移动次数大约是 $M \times N$ 次。下面实验参考代码 2.2 实现了这种简单思想。

为了减少数据的移动次数,第二种方法是通过三次倒序来巧妙地实现。为简单起见,不妨设 $0 \leq M < N$ (否则先进行 $M \% N$ 运算即可),先把 $(A_0 A_1 \cdots A_{N-1})$ 倒序变成 $(A_{N-1} A_{N-2} \cdots A_1 A_0)$,再把它的前 M 个元素 $(A_{N-1} A_{N-2} \cdots A_{N-M})$ 倒序成 $(A_{N-M} \cdots A_{N-1})$,然后把后 $N-M$ 个元素 $(A_{N-M-1} A_{N-M-2} \cdots A_1 A_0)$ 倒序成 $(A_0 A_1 \cdots A_{N-M-1})$ 。这样,整个数组就成了 $(A_{N-M} \cdots A_{N-1} A_0 A_1 \cdots A_{N-M-1})$,这就是我们想要的结果。这种做法每个数据参与 2 次交换(倒序),所以如果一对数据的交换需要 3 次数据移动的话,总共数据移动次数大约是 $3N$ 次。下面实验参考代码 2.3 实现了这种思想。

事实上,还可以有移动次数更少的算法,实验思考题中给出了提示。

(2) 实现要点

M 可以处理成小于 N 的数,以减少移动次数。当 $M > N$ 时,可以用 $M \% N$ 代替 M ,效果相同,但在方法一情况下移动次数大约是 $(M \% N) \times (N+1)$ 次。

方法一:采用循环控制结构。该方法比较简单,只需要“循环右移一位的操作重复进行 M 次”,先定义一个函数进行“每个元素循环右移一位的操作”,然后这个函数被重复调用 M 次。

方法二:采用用异或运算交换数据。该方法比较巧妙,通过三次逆转数组的部分数据就可以实现。程序中定义了一个带参数的宏 `Swap(a,b)`,用连续三次异或运算交换 a 与 b 。当然读者自己也可以通过引入中间变量,通过三次赋值运算实现两个数据的交换。

5. 实验参考代码

方法一:

```
#include <stdio.h>

#define MAXN 100

void Shift( int Array[],int N );

int main()
{
    int Number[MAXN],N,M;
```

```
int i;

scanf("%d%d",&N,&M);
for( i=0; i<N; i++ )
    scanf("%d",&Number[i] );
M %= N; /* 当 M 大于等于 N 时转化成等价的小于 N 的数 */
for( i=0; i<M; i++ )
    Shift(Number,N); /* N 个元素循环位移 1 位 */
for( i=0; i<N-1; i++ ) /* 打印输出 */
    printf("%d ",Number[i]);
printf("%d\n",Number[N-1]);
return 0;
}

void Shift( int Array[],int N )
{
    int i,ArrayEnd;

    ArrayEnd = Array[N-1];
    for( i=N-1; i>0; i-- ) /* N 个元素循环位移 1 位 */
        Array[i] = Array[i-1];
    Array[0] = ArrayEnd;
}
```

代码 2.2 数组元素循环右移的简单解

方法二:

```
#include <stdio.h>

#define MAXN 100
#define Swap(a,b) a ^= b,b ^= a,a ^= b;
/* 通过连续三次异或运算交换 a 与 b */

void RightShift( int Array[],int N,int M );

int main()
{
    // 这里应该写一些测试代码，比如读入 N 和 M，然后调用 RightShift 函数，最后打印输出结果。
}
```

```

int Number[MAXN],N,M;
int i;

scanf("%d%d",&N,&M);
for( i=0; i<N; i++ )
    scanf("%d",&Number[i]);
M %= N; /* 当M大于等于N时转化成等价的小于N的数 */
RightShift(Number,N,M); /* 循环右移M位 */
for( i=0; i<N-1; i++ ) /* 打印输出 */
    printf("%d ",Number[i]);
printf("%d\n",Number[N-1]);
return 0;
}

void RightShift( int Array[],int N,int M )
{
    int i,j;

    if( M>0 && M<N ){
        for(i=0,j=N-1; i<j; i++,j--) /* 逆转N个数据 */
            Swap(Array[i],Array[j]);
        for(i=0,j=M-1; i<j; i++,j--) /* 逆转前M个数据 */
            Swap(Array[i],Array[j]);
        for(i=M,j=N-1; i<j; i++,j--) /* 逆转后N-M个数据 */
            Swap(Array[i],Array[j]);
    }
}

```

代码 2.3 数组元素循环右移的 $3N$ 次位移解

6. 实验思考题

(1) 如果修改题目要求,允许使用另外一个大小为 M (假设 $N > M > 0$) 的数组,则如何提高程序效率?

(2) 不改变题目的任何限定,能否设计一种移动次数不超过 $2N$ 的方法? [提示:可以通过分析每个数据原位置与目标位置之间的下标关系,将每个数据一次性定位。根据题目的要求,可以发现:任何位于数组下标 i 位置的数据,其目的地址是下标为 $(i + M) \% N$ 的位置,或者说第