

一流学校 一流老师 一流资源



三一丛书

数据结构

要点与解题

朱战立 张选平 编著



西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

西安交大教学资源文库 三一丛书

数据结构

要点与解题

朱战立 张选平 编著

西安交通大学出版社

内 容 提 要

计算机是广泛应用的工具。数据结构课程是学习计算机软件设计的基础课程。本书是作者在长期教学经验积累的基础上精心编著的数据结构课程的学习参考书。

本书共分 10 章,包括线性表、堆栈、队列、数组、串、广义表、树、二叉树、图、排序、查找等典型数据结构内容。每章主要由基本内容和典型题解两部分组成。基本内容部分简述了该章的学习要求,以及基本术语和基本概念,是该章学习和考试复习的大纲。典型题解部分的例题主要分为概念题和算法设计题。这些例题都是作者精心编选的,具有典型意义。有些例题后边还有对类似问题的深入讨论。为辅导学生完成上机作业题,大部分章中都包含了上机实习题解部分。本书的算法用 C 语言描述。本书的所有算法和上机实习题程序都在计算机上调试通过。

本书是计算机专业本科学生、报考计算机专业硕士研究生的考生的非常适宜的学习参考书。

图书在版编目(CIP)数据

数据结构要点与解题/朱战立,张选平编著. —西安:
西安交通大学出版社,2006.8
(西安交大教学资源文库.“三一”丛书)
ISBN 7-5605-2251-3

I. 数… II. ①朱…②张… III. 数据结构-高等
学校-教学参考资料 IV. TP311.12

中国版本图书馆 CIP 数据核字(2006)第 075510 号

书 名:数据结构要点与解题

编 著:朱战立 张选平

出版发行:西安交通大学出版社

地 址:西安市兴庆南路 25 号(邮编:710049)

电 话:(029)82668357 82667874(发行部)

(029)82668315 82669096(总编办)

印 刷:陕西向阳印务有限公司

字 数:287 千字

开 本:880 mm×1 230 mm 1/32

印 张:7.875

版 次:2006 年 8 月第 1 版 2006 年 9 月第 2 次印刷

书 号:ISBN 7-5605-2251-3/TP·443

定 价:12.80 元

版权所有 侵权必究

丛书总序

为了使普通高等学校理工类专业的大学生更好地学习、掌握基础课和专业基础课知识,我们组织出版了这套“三一”丛书,目的就是提供一流的学习资源,使大家共享一流教师的教学经验和教学成果,为今后的学习打下良好的基础。

西安交通大学是国内仅有的几所具有百年历史的高等学府,是首批进入国家“211 工程”建设的七所大学之一,1999 年被国家确定为我国中西部地区惟一所以建设世界知名高水平大学为目标的学校。西安交大历来重视本科生教学,1996 年成为全国首家本科教学评估为优秀的大学。学校拥有国家级、省部级、校级教学名师数十名,具有丰富的、一流的教学资源。本丛书均由西安交通大学长期在教学一线主讲的教授、副教授主编,他们具有丰富的基础课、专业基础课教学和辅导经验。丛书作者们在长期的教学实践中,深深了解学生在学习基础课、专业基础课时的难点和困惑点之所在,对如何使学生更有效地学习、掌握课程的基本知识和解题技巧进行了深入的探索和研究,并将成果体现于书中。

本丛书针对中少学时课程的特点和教学要求,以普通高等学校的学生为主要对象,不拘泥于某一本教材,而是将有特色和使用量较大的各种版本的教材加以归纳总结,取其精华,自成一体。书中对课程的基本内容、研究对象、教学要求、学习方法、解题思路进行了全面、系统的总结和提炼,按基本知识点、重点与难点、典型题解析、自我检测题等环

节进行编排。本丛书既可单独使用,也可与其他教材配合使用。

我们衷心希望本丛书成为您大学基础课和专业基础课学习阶段的良师益友,帮助您克服困难,进入大学学习的自由王国,并祝您早日成为国家的栋梁之材!

在学习使用过程中,您如果发现书中有不妥之处或有好的建议,敬请批评指正并反馈给我们,我们会进一步改进自己的工作,力争使您满意。

真诚感谢您使用西安交大版图书。

西安交大出版社网址:<http://press.xjtu.edu.cn>

<http://www.xjtupress.com>

理工医事业部信箱: jdlgy31@126.com

西安交通大学出版社

2006年6月

前 言

数据结构是计算机学科的一门核心课,也是其它一些和计算机学科关系密切学科或专业的一门必修课或选修课。计算机是广泛应用的工具。数据结构课程是学习计算机软件设计的基础课程。

本书作者长年从事数据结构课程的教学,出版过多本在全国有广泛影响的数据结构教科书。本书是作者在长期教学经验积累的基础上精心编著的数据结构课程的学习参考书。

本书共分10章,包括线性表、堆栈、队列、数组、串、广义表、树、二叉树、图、排序、查找等典型数据结构课程内容。每章主要由基本内容和典型题解两部分组成。为辅导学生完成上机作业题,一些章中还包含了上机实习题解部分。基本内容部分简述了该章的学习要求,以及基本术语和基本概念,是该章学习和考试复习的大纲。典型题解部分的例题主要分为概念题和算法设计题。这些例题都是作者精心编选的,具有典型意义。作者在解题时注意结合基本概念进行分析,对读者理解学习内容和掌握解题方法会有帮助。有些例题后边还有对类似问题的深入讨论。数据结构课程包含上机实习,许多学生在完成上机实习题时感觉非常困难,上机实习题解部分是专为解决这些问题设计的。

本书的算法用C语言描述。本书的所有算法和上机实习题程序都在计算机上调试通过。

本书是计算机专业本科学生、报考计算机专业硕士研究生的考生的非常适宜的学习参考书。

本书的第7,8,9章由张选平编著,其余各章由朱战立编著,全书由

朱战立修改定稿。

尽管作者在写作过程中非常认真和努力,但由于水平有限,错误和不足之处在所难免,敬请读者批评指正。

编著者

2006.5

目 录

丛书总序

前 言

第 1 章 绪论

1.1 基本内容	(1)
1.1.1 学习要求	(1)
1.1.2 基本概念	(1)
1.2 典型题解	(2)
1.2.1 概念题	(2)
1.2.2 算法分析题	(4)
1.2.3 算法设计题	(6)

第 2 章 顺序存储结构的表、堆栈和队列

2.1 基本内容	(8)
2.1.1 学习要求	(8)
2.1.2 基本概念	(8)
2.1.3 基本结构体定义	(9)
2.2 典型题解	(9)
2.2.1 顺序表及其应用	(9)
2.2.2 堆栈、顺序堆栈及其应用	(13)
2.2.3 顺序循环队列及其应用	(16)
2.2.4 顺序双向循环队列及其应用	(20)
2.2.5 顺序优先级队列及其应用	(21)
2.3 上机实习典型题解	(23)

第 3 章 链式存储结构的表、堆栈和队列

3.1 基本内容	(26)
3.1.1 学习要求	(26)

3.1.2	基本概念	(26)
3.1.3	基本结构体定义	(27)
3.2	典型题解	(27)
3.2.1	带头结点和不带头结点的单链表及其应用	(27)
3.2.2	链式堆栈及其应用	(37)
3.2.3	尾指针表示的单循环链表及其应用	(38)
3.2.4	尾指针表示的链式队列及其应用	(40)
3.3	上机实习典型题解	(42)
第4章 串、数组和矩阵		
4.1	基本内容	(49)
4.1.1	学习要求	(49)
4.1.2	基本概念	(49)
4.1.3	基本结构体定义	(50)
4.2	典型题解	(51)
4.2.1	串的基本概念和应用问题	(51)
4.2.2	数组的基本概念和应用问题	(58)
4.2.3	特殊矩阵和稀疏矩阵的基本概念和应用问题	(59)
4.3	上机实习典型题解	(68)
第5章 递归程序设计		
5.1	基本内容	(73)
5.1.1	学习要求	(73)
5.1.2	基本概念	(73)
5.2	典型题解	(74)
5.2.1	基本的递归概念和递归算法执行过程	(74)
5.2.2	复杂的递归概念和应用问题	(80)
5.3	上机实习典型题解	(89)
第6章 广义表		
6.1	基本内容	(93)
6.1.1	学习要求	(93)
6.1.2	基本概念	(93)
6.1.3	存储结构	(94)
6.2	典型题解	(96)

6.2.1	基本概念题	(96)
6.2.2	算法设计题	(98)
第7章 树与二叉树		
7.1	基本内容	(106)
7.1.1	学习要求	(106)
7.1.2	基本概念	(106)
7.1.3	基本结构体定义	(107)
7.2	典型题解	(108)
7.2.1	基本内容题	(108)
7.2.2	算法设计与分析题	(117)
7.3	上机实习典型题解	(135)
第8章 图		
8.1	基本内容	(140)
8.1.1	学习要求	(140)
8.1.2	基本概念	(140)
8.1.3	基本结构体定义	(142)
8.2	典型题解	(143)
8.2.1	基本内容题	(143)
8.2.2	算法设计与分析题	(152)
8.3	上机实习典型题解	(168)
第9章 排序		
9.1	基本内容	(175)
9.1.1	学习要求	(175)
9.1.2	基本概念	(175)
9.1.3	主要结构体定义	(177)
9.2	典型题解	(177)
9.2.1	基本内容题	(177)
9.2.2	算法设计与分析题	(184)
9.3	上机实习典型题解	(200)
第10章 查找		
10.1	基本内容	(209)
10.1.1	学习要求	(209)

10.1.2	基本概念	(209)
10.2	典型题解	(211)
10.2.1	线性表查找题解	(211)
10.2.2	树型表查找题解	(211)
10.2.3	哈希表查找题解	(222)
10.3	上机实习典型题解	(225)
附录 1	算法书写规范	(231)
附录 2	上机实习内容规范	(232)
附录 3	西安某高校 2005 年硕士研究生入学考试试题和解答	(233)
附录 4	西安某高校 2006 年硕士研究生入学考试试题和解答	(237)

第 1 章 绪 论

1.1 基本内容

1.1.1 学习要求

本章基本知识点:

- ★ 数据结构的基本概念
- ★ 数据结构的基本术语
- ★ 数据类型和抽象数据类型
- ★ 算法和算法设计目标
- ★ 算法效率度量

本章重点:

- ★ 数据结构的基本概念
- ★ 算法分析入门

本章难点:

- ★ 基本概念和基本术语
- ★ 算法分析入门

1.1.2 基本概念

数据结构: 一门研究非数值计算的程序设计问题中计算机的操作对象以及它们之间关系和操作的课程。数据结构被形式地定义为 (D, R) , 其中 D 是数据元素的有限集合, R 是 D 上的关系。

数据: 人们利用文字符号、数字符号以及其他规定的符号对现实世界的事物及其活动所做的抽象描述。

数据元素: 表示数据的基本单位。数据元素通常由若干个数据项组成。

数据的逻辑结构: 数据元素之间的联系结构。

数据的存储结构: 数据的逻辑结构在计算机中的存储方式。也称数据的物理结构。数据存储结构的基本形式有两种: 一种是顺序存储结构, 另一种是链式存

储结构。

顺序存储结构:把数据元素存储在一块连续地址空间的内存中,其特点是逻辑上相邻的数据元素在物理上也相邻,数据间的逻辑关系表现在数据元素的存储位置关系上。

链式存储结构:使用指针把发生联系的数据元素链接起来,其特点是逻辑上相邻的数据元素在物理上不一定相邻,数据间的逻辑关系表现在结点的链接关系上。

数据的操作:允许对数据元素进行的操作(或称运算)。

类型:一组值的集合。例如,整数类型通常是 $-32768 \sim 32767$ 。

数据类型:指一个类型和定义在这个类型上的操作集合。例如,int 类型是集合 $[-32768, 32767]$ 和定义在这个集合上的加(+)、减(-)、乘(*)、除(/)和求模(%)运算。

抽象数据类型:指一个逻辑概念上的类型和这个类型上的操作集合。例如,圆的抽象数据类型包括描述圆的基本数据(圆的半径)和常规的运算(如求圆的面积、体积等)。

算法:对特定问题求解步骤描述的计算机指令的有限序列。

算法需满足的性质:输入性、输出性、有限性、确定性、可执行性。

算法的时间复杂度:算法执行时间和数据元素个数 n 之间的函数关系。也称作算法的时间效率。通常采用 $O(f(n))$ 表示法。

算法的空间复杂度:执行算法所需内存空间和数据元素个数 n 之间的函数关系。也称作算法的空间效率。通常采用 $O(f(n))$ 表示法。

1.2 典型题解

1.2.1 概念题

例 1-1 填空题

- (1)数据结构是一门研究非数值计算的程序设计问题中计算机的()以及它们之间()和操作的课程。
- (2)数据结构被形式地定义为 (D, R) ,其中 D 是()的有限集合, R 是 D 上的()。
- (3)数据结构包括数据的()、数据的()和数据的()这三个方面的内容。
- (4)数据结构按逻辑结构可分为两大类,它们分别是()和()。

(5)数据的操作最常用的有5种,分别是()、()、()、()和()。

(6)算法的效率可分为()效率和()效率。

答:

(1)操作对象,关系

(2)数据元素,关系

(3)逻辑结构、存储结构、操作

(4)线性结构、非线性结构

(5)插入、删除、修改、查找、排序

(6)时间、空间

例 1-2 选择题

(1)非线性结构是数据元素之间存在一种

A. 一对多关系

B. 多对多关系

C. 多对一关系

D. 一对一关系

(2)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

(3)算法分析的目的是

A. 找出数据结构的合理性

B. 研究算法中的输入和输出的关系

C. 分析算法的效率以求改进

D. 分析算法的易懂性和文档性

(4)算法分析的两个主要方面是

A. 时间复杂性和空间复杂性

B. 正确性和简明性

C. 可读性和文档性

D. 数据复杂性和程序复杂性

(5)算法指的是

A. 计算方法

B. 排序方法

C. 解决问题的有限运算序列

D. 调度方法

(6)计算机算法必须具备输入、输出和_____等5个特性。

A. 可行性、可移植性和可扩充性 B. 有限性、确定性、可执行性

C. 确定性、有穷性和稳定性

D. 易读性、稳定性和安全性

答:

(1) B (2) C (3) C (4) A (5) C (6) B

例 1-3 简答题

(1)数据结构和数据类型两个概念之间有区别吗?

(2)简述线性结构与非线性结构的不同点。

答:

(1)简单地讲,数据结构定义了一组按某些关系结合在一起的数组元素。数据类型不仅定义了一组带结构的数据元素,而且还在其上定义了一组操作。

(2)线性结构反映数据元素间的逻辑关系是一对一的,非线性结构反映数据元素间的逻辑关系是多对多的。

1.2.2 算法分析题

例 1-4 设数组 a 和 b 在前边部分已赋值,求如下两个 n 阶矩阵相乘运算程序段的时间复杂度。

```
for(i = 0; i < n; i++)
    for(j = 0; j < n; j++)
    {
        c[i][j] = 0; /* 基本语句 1 */
        for(k = 0; k < n; k++)
            c[i][j] = c[i][j] + a[i][k] * b[k][j];
        /* 基本语句 2 */
    }
```

答:设基本语句的执行次数为 $f(n)$,有

$$f(n) = n^2 + n^3$$

因程序段的时间复杂度 $T(n) = f(n) = n^2 + n^3 \leq c * n^3 = O(n^3)$,其中 c 为常数,所以该程序段的时间复杂度为 $O(n^3)$ 。

例 1-5 设 n 为如下程序段处理的数据个数,求如下程序段的时间复杂度。

```
for(i = 1; i <= n; i = 2 * i)
    printf("i = %d\n", i); /* 基本语句 */
```

答:设基本语句的执行次数为 $f(n)$,有 $2f(n) \leq n$, 即有 $f(n) \leq 1bn$

因程序段的时间复杂度 $T(n) = f(n) \leq 1bn \leq c * 1bn = O(1bn)$,其中 c 为常数,所以该程序段的时间复杂度为 $O(1bn)$ 。

在很多情况中,算法中数据元素的取值情况不同算法的时间复杂度也会不同。此时算法的时间复杂度应是数据元素最坏情况下取值的时间复杂度或数据元素等概率取值情况下的平均(或称期望)时间复杂度。

例 1-6 下边的算法是用冒泡排序法对数组 a 中的 n 个整型的数据元素($a[0] \sim a[n-1]$)从小到大排序的算法,求该算法的时间复杂度。

```
void BubbleSort(int a[], int n)
{
```

```

int i, j, flag=1;
int temp;

for(i = 1; i < n && flag == 1; i++)
{
    flag = 0;
    for(j = 0; j < n-i; j++)
    {
        if(a[j] > a[j+1])
        {
            flag = 1;
            temp = a[j];
            a[j] = a[j+1];
            a[j+1] = temp;
        }
    }
}

```

答:这个算法的时间复杂度随待排序数据的不同而不同。当某次排序过程中没有任何两个数组元素交换位置,则表明数组元素已排序完毕,此时算法将因标记 $\text{flag}=0$ 不满足循环条件而结束。但是,在最坏情况下,每次排序过程中都至少有两个数组元素交换位置,因此,应按最坏情况计算该算法的时间复杂度。

设基本语句的执行次数为 $f(n)$,最坏情况下有:

$$f(n) \approx n + 4 * n^2 / 2$$

因算法的时间复杂度 $T(n) = f(n) \approx n + 2 * n^2 \leq c * n^2 = O(n^2)$,其中 c 为常数,所以该算法的时间复杂度为 $O(n^2)$ 。

例 1-7 下边算法是在一个有 n 个数据元素的数组 a 中删除第 i 个位置的数组元素,要求当删除成功时数组元素个数减 1,求该算法的时间复杂度。其中,数组下标从 0 至 $n-1$ 。

```

int Delete(int a[], int *n, int i)
{
    int j;
    if(i < 0 || i > *n) return 0;          /* 删除位置错误,失败返回 */
    for(j = i + 1; j < *n; j++) a[j-1] = a[j]; /* 顺次移位填补 */
}

```



```

    (*n)--; /* 数组元素个数减 1 */
    return 1; /* 删除成功返回 */
}

```

答:这个算法的时间复杂度随删除数据的位置不同而不同。当删除最后一个位置的数组元素时有 $i = n-1, j = i+1 = n$, 此时因不需移位填补而循环次数为 0, 当删除倒数最后一个位置的数组元素时有 $i = n-2, j = i+1 = n-1$, 此时因只需移位填补一次而循环次数为 1, 依此类推, 当删除第一个位置的数组元素时有 $i = 0, j = i+1 = 1$, 此时因需移位填补 $n-1$ 次而循环次数为 $n-1$ 。此时算法的时间复杂度应是删除数据的位置等概率取值情况下的平均时间复杂度。

假设删除任何位置上的数据元素都是等概率的(一般情况下均可做等概率假设), 设 P_i 为删除第 i 个位置上数据元素的概率, 则有 $P_i = 1/n$, 设 E 为删除数组元素的平均次数, 则有:

$$\begin{aligned}
 E &= \frac{1}{n} \sum_{i=0}^{n-1} (n-1-i) = \frac{1}{n} [(n-1) + (n-2) + \dots + 2 + 1 + 0] \\
 &= \frac{1}{n} \cdot \frac{n(n-1)}{2} = \frac{n-1}{2}
 \end{aligned}$$

因该算法的时间复杂度 $T(n) = E \leq (n+1)/2 \leq c * n = O(n)$, 其中 c 为常数, 所以该算法的时间复杂度为 $O(n)$ 。

1.2.3 算法设计题

例 1-8 设计一个从两个整数类型数据中得到较大数值的算法。

算法设计如下:

```

int Max1(int x1, int x2)
{
    if(x1 >= x2) return x1;
    else return x2;
}

```

说明:在上述算法的设计中,函数名具有整数类型的返回值,用于返回得到的较大数值;两个参数均为输入参数(即值参),参数的类型均为整数类型。

例 1-9 设计一个从三个整数类型数据中得到最大数值和次大数值的算法。

设计分析:算法要有三个输入参数,分别表示三个整数类型数据;由于算法要带回两个返回值,而函数名只能带回一个返回值,因此必须设计两个输出参数,用于带回算法的两个结果值,其类型为整数的指针类型。

算法设计如下: