

# 工科课程**提高**与**应试**丛书

- 涵盖课程重点及难点
- 精设典型题详解及评注
- 选配课程考试模拟及全真试卷

赵仲孟 张 蓓 编

# 数据结构

## 典型题解析及自测试题



西北工业大学出版社

工科课程提高与应试丛书

数 据 结 构  
典型题解析及自测试题

赵仲孟 张 蓓 编

西北工业大学出版社

**【内容简介】** 本书是根据数据结构课程的教学大纲的要求并以编者多年教学中积累和收集的有关资料、题解编写的。本书分两部分,第一部分每章按内容提要、典型题解析和习题3个模块组成;第二部分有4套自测试题。附录中给出习题和自测试题的答案。本书旨在帮助学生熟练掌握数据结构课程的基本理论知识、重点、难点和解题的方法与技巧,概念清晰,习题覆盖面广。

本书可作为计算机专业本、专科学生的学习参考书,也是报考计算机专业硕士研究生的考生必读参考书,并适用于自学考试和计算机等级考试的读者。

## 图书在版编目(CIP)数据

数据结构典型题解析及自测试题/赵仲孟,张蓓. —西安:西北工业大学出版社,2002.2  
(工科课程提高与应试丛书)  
ISBN 7-5612-1467-7

I. 数… II. ①赵… 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(2002)第 021041 号

出版发行:西北工业大学出版社

通讯地址:西安市友谊西路127号 邮编:710072 电话:029-8493844

网 址: <http://www.nwpu.com>

印刷者:西北工业大学出版社印刷厂

开 本:850 mm×1 168 mm 1/32

印 张:9.25

字 数:234千字

版 次:2002年8月第1版 2002年8月第1次印刷

印 数:1~5 000册

定 价:13.00元

# 前 言

数据结构是研究数据以及数据之间关系的一门学科。国外有本书的名字叫《程序=数据结构+算法》，这正是数据结构课程所要研究的内容。计算机编程中加工处理的对象是数据，而数据具有一定的组织结构，所以学习计算机编程，仅仅了解计算机语言是不够的，还必须掌握数据组织、存储和运算的一般方法，这也是我们编写计算机程序的重要基础。由于它对计算机学科起着承前启后的作用，因此，本课程被列为计算机等相关专业主要的专业基础课程之一。

本书是按照数据结构课程的教学大纲要求，根据编者长期从事本科生教学和研究生考前辅导的教学研究和教学改革实践经验编写的辅导教材。目的就是通过对课程要点复习和习题解答，使读者充分掌握数据结构的原理以及求解数据结构问题的思路与方法，深化对基本概念的理解，提高分析与解决问题的能力。

本书内容涵盖数据结构课程的教学大纲的要求，书中第一部分按内容分为8章，每章分内容提要、典型题解析和习题3个模块。内容提要是属于课程要求理解和掌握的内容，编者根据自己多年的教学经验，总结了课程的核心内容，以帮助读者更好地理解掌握基本理论、知识和技能。典型题解析是按章选择具有代表性的典型题目，在解题之前给出了题目“分析”，指出解题的思路、方法和步骤，解题之后又给出题目的相应“评注”，主要强调该题目须要注意的问题；习题中有选择判断、分析解答，力求使学生接触到不同风格、类型的题目，以帮助学生理解、掌握数据结构课程基本理论知识，顺利完成课程学习和拓展知识面。第

二部分给出了 4 套自测试题。附录中给出了习题和自测试题答案，供学生复习完本课程后，进行自我检测学习效果。

本书习题覆盖面广，既收集了较容易的题目，也收集了难度适中和较高难度的题目。因此，本书不仅可以作为计算机专业本、专科生学习数据结构课程的参考书，也是报考计算机专业硕士研究生的考生首选参考书，还适用于自学考试和计算机等级考试的读者。

全书由赵仲孟主编并统稿。其中第一至八章的内容提要由赵仲孟编写，第一至五章的习题部分和全部自测试题由张蓓编写，第六章习题部分由霍秋艳编写，第七章习题部分由刘君瑞编写，第八章习题部分由沈海斌和王瑞编写。张选平副教授为本书的习题编写提供了大量的宝贵意见，西安交大电信学院软件所同仁特别是数据结构教学小组的老师们对本书的出版给予了大力支持和帮助，在此一并致谢。

由于时间仓促，书中习题解答上可能存在不够完整和疏漏之处，恳请读者不吝赐教。

编 者

2001 年 9 月于西安交大

# 目 录

## 第一部分 典型题解析

|            |    |
|------------|----|
| 第一章 引 论    | 1  |
| 一、内容提要     | 1  |
| 二、典型题解析    | 2  |
| 三、习题       | 5  |
| 第二章 线性表    | 8  |
| 一、内容提要     | 8  |
| 二、典型题解析    | 13 |
| 三、习题       | 39 |
| 第三章 栈和队列   | 42 |
| 一、内容提要     | 42 |
| 二、典型题解析    | 44 |
| 三、习题       | 60 |
| 第四章 数组与广义表 | 63 |
| 一、内容提要     | 63 |
| 二、典型题解析    | 64 |
| 三、习题       | 83 |

|                          |            |
|--------------------------|------------|
| <b>第五章 树和二叉树 .....</b>   | <b>85</b>  |
| 一、内容提要 .....             | 85         |
| 二、典型题解析 .....            | 89         |
| 三、习题 .....               | 132        |
| <b>第六章 图 .....</b>       | <b>136</b> |
| 一、内容提要 .....             | 136        |
| 二、典型题解析 .....            | 140        |
| 三、习题 .....               | 180        |
| <b>第七章 查 找 .....</b>     | <b>185</b> |
| 一、内容提要 .....             | 185        |
| 二、典型题解析 .....            | 190        |
| 三、习题 .....               | 202        |
| <b>第八章 内排序和外排序 .....</b> | <b>207</b> |
| 一、内容提要 .....             | 207        |
| 二、典型题解析 .....            | 211        |
| 三、习题 .....               | 229        |

## 第二部分 自测试题

|             |     |
|-------------|-----|
| 自测试题一 ..... | 233 |
| 自测试题二 ..... | 236 |
| 自测试题三 ..... | 238 |
| 自测试题四 ..... | 241 |

## 附录 习题与自测试题答案

|     |             |     |
|-----|-------------|-----|
| 附录一 | 习题答案·····   | 245 |
| 附录二 | 自测试题答案····· | 278 |

# 第一部分 典型题解析

## 第一章 引 论

### 一、内容提要

#### (一) 数据结构的概念

数据结构 DS 是研究数据及数据之间的关系的一门学科。

Data → Data Element → Data Structure

DS { (1) 逻辑结构: 定义  $DS = (D, S)$   
                     $D$  为元素集,  $S$  为关系集。  
(2) 存储结构: 数据及其关系的计算机表示。  
(3) 数据的运算: 定义在数据逻辑结构上的基本操作  
                    运算的实现依赖于存储结构。

Data → 二进制表示的符号的总称

Data Element → 体现 C 语言中的各种数据类型, 如基本类型, 结构、数组、联合、文件等。

#### (二) 数据之间的关系

(1) 逻辑结构描述了数据元素之间的关系, 与计算机的具体

实现无关。逻辑结构分为两大类：① 线性结构；② 非线性结构（其中树形结构和图状结构是两种典型的非线性结构）。

（2）物理结构是逻辑结构在计算机中的实现。物理结构分为顺序存储结构和链式存储结构。

### （三）数据的算法

（1）算法的概念：算法是一个有穷的规则序列。它决定了解决某一特定问题的一系列运算。即算法是一个指令的有限集合。

（2）算法特点：有穷性、确定性、可行性、输入、输出。

（3）算法评价：在“正确地”前提下，评价有两个指标：时/空。

1) 时间复杂度：算法程序运行时的消耗时间，用问题规模的数量级表述为

$$T(n) = O(f(n)) \quad \text{如：} \quad T(n) = O(n^2)$$

2) 空间复杂度：算法程序运行时所占用的存储大小。

$$S(n) = O(f(n)) \quad \text{如：} \quad S(n) = O(n)$$

## 二、典型题解析

**例 1.1** 设  $n$  为正整数，确定下列划线语句的执行频度。

(1) for( $i=1; i \leq n; i++$ )

for( $j=1; j \leq i; j++$ )

for( $k=1; k \leq j; k++$ )

$x += \text{deltn};$

(2)  $x=91; y=100;$

while( $y>0$ )

{

if( $x>100$ )

{

```

        x--=10; y----;
    }
    else
        x++;
}
(3) i=1;
    while(i<=n)
        i=i*3;
(4) for(i=1; i<=n; i++)
{
    k=i+1;
    do
    {
        k=k+1;
    } while(k<=n)
}

```

**分析** 语句的执行频度是该语句重复执行的次数。计算循环语句段中某一语句的执行次数,要得到语句执行与循环变量之间的关系。

**解** (1) 这是一个三层嵌套循环,最内层的循环次数由  $j$  决定,次内层的循环次数由  $i$  决定,而  $i$  从 1 变化到  $n$ 。

所以划线语句的执行频度为

$$\sum_{i=1}^n \sum_{j=1}^i j$$

(2) 在程序中,  $x$  的初始值为 91, 当  $x$  小于 100 时,  $\{x--=10; y----; \}$  语句不会执行, 只有当  $x$  增长到 101 时,  $\{x--=10; y----; \}$  语句才能执行。  $x$  从 91 增长到 101, 循环须要执行 10 次。 在第 11 次循环时,  $x=101$ ,  $\{x--=10; y----; \}$  语句的执行结果使  $x$  又

恢复为原值 91, 同时  $y$  减 1。所以  $y$  每减少 1, while 循环就须要执行 11 次, 而划线语句执行了 10 次。所以划线语句的执行频度为  $100 \times 10$ 。

(3) 设语句执行  $m$  次, 则有

$$3^m \leq n \Rightarrow m = \log_3 n$$

(4) 变量  $i$  从 1 变化到  $n$ ,  $k$  从 2 变化到  $n+1$ 。当  $k = i+1 (i < n)$  时, 内部 do-while 循环执行  $(n+1-k)$  次, 而  $k = n+1$  时, 仍执行一次。所以划线语句的执行次数为

$$\sum_{k=2}^n (n+1-k) + 1 = \sum_{i=1}^{n-1} (n-i) + 1$$

**例 1.2** 增长率由小到大的顺序排列下列各函数:

$2^{100}$ ,  $\left(\frac{3}{2}\right)^n$ ,  $\left(\frac{2}{3}\right)^n$ ,  $n^n$ ,  $n^{3/2}$ ,  $n^{2/3}$ ,  $\sqrt{n}$ ,  $n!$ ,  $n$ ,  $\log_2 n$ ,  $n \log_2 n$ ,  $n^{\log_2 n}$

**解** 各函数的排列顺序如下:

$\left(\frac{2}{3}\right)^n$ ,  $2^{100}$ ,  $\log_2 n$ ,  $\sqrt{n}$ ,  $n^{2/3}$ ,  $n$ ,  $n \log_2 n$ ,  $n^{3/2}$ ,  $\left(\frac{3}{2}\right)^n$ ,  $n^{\log_2 n}$ ,  $n!$ ,  $n^n$

**例 1.3** 分析下面的递归函数的时间复杂度。

```
int func(int n)
{
    if (n <= 1) return (1);
    else return (func(n-1) * n);
}
```

(1)

(2)

**分析** 设  $\text{func}(n)$  的运行时间为  $T(n)$ , 由程序可知

$$T(n) = \begin{cases} O(1) & n \leq 1 \\ T(n-1) + O(1) & n > 1 \end{cases}$$

所以

$$\begin{aligned} T(n) &= O(1) + T(n-1) = 2 \times O(1) + T(n-2) = \\ &\cdots = n \times O(1) = O(n) \end{aligned}$$

解 时间复杂度为  $O(n)$ 。

例 1.4 算法 A 的每次循环需要 200 个机器周期, 解决一个问题规模为  $n$  的问题时需要  $n \log n$  次循环。另一个简单的算法 B 每次循环需要 25 个机器周期, 解决问题规模为  $n$  的问题时需要  $n$  的平方次循环。在什么情况下, 会选择算法 A 而不是算法 B?

分析 在解决某个问题时, 算法需要的总机器周期与问题规模  $n$  有关。只要了解当  $n$  在什么情况下, 算法 A 使用的总机器周期比算法 B 使用的总机器周期小, 就可得到此题问题的答案。

例 1.5 判断: 在一个含有  $n$  个元素的数组中是否含有判断一对不相等的元素的算法, 其时间复杂度小于  $\Omega(n \log n)$ 。

分析 算法可以是线性级的, 只需要简单地将第一个元素与其他  $n-1$  个元素相比较。

例 1.6 选择解决某种问题的最佳数据结构的标准是什么?

分析 通常有两条标准: 一是所需的存储空间量; 二是算法所需要的时间。

对于算法所需要的时间又涉及以下几点:

- (1) 程序运行时所需输入的数据总量;
- (2) 对源程序编译所需的时间;
- (3) 计算机执行每条指令所需的时间;
- (4) 程序中的指令重复执行的次数。

### 三、习 题

#### 一、选择题

1. 在数据结构中, 从逻辑上可以把数据结构分成( )。

- 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
2. 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( )  
及它们之间的( )和运算等的学科。
- 1) A. 操作对象      B. 计算方法  
C. 逻辑存储      D. 存储映像
- 2) A. 结构      B. 关系  
C. 运算      D. 算法
3. 每种数据结构的逻辑结构与物理结构总是一致的, 此种说法( )。
- A. 正确      B. 不正确
4. 每种数据结构都具备三个基本运算: 插入、删除和查找, 这种说法( )。
- A. 正确      B. 不正确

## 二、指出下列各算法的时间复杂度

- (1)    void prime(int n)  
      {  
          int i=2;  
          while ((n%i) != 0 && i \* 1.0 < sqrt(n)) i++;  
          if(i \* 1.0 > sqrt(n))  
              printf("%d is a prime \n", n);  
          else  
              printf("%d is not a prime \n", n);  
      }
- (2)    void sum(int n)  
      {  
          int p=1, sum=0, i;  
          for (i=1; i < n; i++)  
          {  
              p \*= i;

```
        sum += p;
    }
    return sum;
}

(3) void sum2(int n)
{
    int sum=0,i,j;
    for (i=1;i<n;i++)
    {
        p=1;
        for (j=1;j<i;j++) p*=j;
        sum+=p;
    }
    return sum;
}
```

## 第二章 线 性 表

### 一、内容提要

#### (一) 线性表的定义

线性表是  $n$  个结点的有限序列  $(a_1, a_2, \dots, a_n)$ 。对非空表而言,满足以下 4 条:

- (1) 有一个头结点,头结点无前驱( $a_1$ );
- (2) 有最后一个结点,尾结点无后续( $a_n$ );
- (3)  $a_k (1 < k < n)$  在  $a_{k-1}$  之后,即  $a_k$  为  $a_{k-1}$  的后续;
- (4)  $a_k$  在  $a_{k+1}$  之前,即  $a_k$  为  $a_{k+1}$  的前驱。

结点、数据元素、节点、顶点等均为一个意思,即处理对象。其中结点是数据元素的物理映像。

#### (二) 线性表的物理存储结构

##### 1. 顺序存储

(1) 定义:结点按顺序存放在一块连续的存储单元中。

(2) 存储特点:逻辑顺序与物理顺序一致,线性关系利用物理上的相邻关系来体现。

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

式中, $L$  为结点大小。

(3) 顺序存储上的操作:

- 1) 初始化:预先分配空间。

2) 插入操作:要判断空间是否占满,如果没有可用空间,要重新分配空间;需要将插入位置之后的元素后移。时间复杂度为  $O(n)$ 。

3) 删除操作:须将删除元素之后的元素前移。时间复杂度为  $O(n)$ 。

(4) 在顺序存储结构上实现线性表的优缺点:

优点:1) 逻辑结构与物理结构一致;

2) 随机存取元素(存取元素时间相等),即元素定位公式化;

3) 简单、直观;

缺点:1) 插入、删除时要移动大量元素;

2) 扩展不灵活;

3) 长度过大时,预留空间浪费。

## 2. 链式存储

(1) 定义:表中的结点包括两个域:数据域和指针域,线性关系通过指针来实现。

包括单链表、循环链表、双向链表、双向循环链表。

(2) 存储特点:元素的物理位置不再反映元素之间的逻辑关系,而是用指针来反映元素之间的逻辑关系。

(3) 在链式存储结构上实现线性表的优缺点:

优点:1) 动态结构,无须预分配空间。可用任意的(连续或者不连续)存储单元存放元素。

2) 操作灵活。

缺点:1) 非随机存储结构;

(2) 指针操作复杂。

• **线性单链表** 分为 不带头结点的单链表、带头结点的单链表和静态链表三种。

注意:静态链表是用一维数组结构来模拟链表,其物理的相邻