数据结构

考研指导

计算机专业考研系列教材

李春葆 编著

清华大学出版社

计算机专业考研系列教材

数据结构考研指导

李春葆　　编著

清 华 大 学 出 版 社

（京）新登字 158 号

内 容 简 介

数据结构是计算机及其相关专业的核心课程，也是绝大多数高校招收计算机专业硕士研究生的必考科目之一。本书由长期坚持在教学一线的教授亲自主笔。书中融汇了数据结构这门课程的特点、难点、知识点和考研的出题重点，提供了丰富的例题和练习题，包含了大量的研究生入学试题。

本书共分 11 章。第 1 章是数据结构的概述；第 2 章~第 11 章以每章一个专题的形式，分别讨论了线性表、栈和队列、串、数组和广义表、树和二叉树、图、查找、内排序、外排序和文件等基本类型的数据结构。各章均由核心考点、例题分析、基础要点总结、练习题及参考答案 4 部分组成。其中的例题分析部分强调解题思路，注重算法分析。

本书主要针对计算机及相关专业硕士研究生入学考试，也可作为计算机类专业或信息类专业的本科或专科教材，还可供从事计算机工程与应用工作的科技工作者参考。

图书在版编目（CIP）数据

数据结构考研指导 / 李春葆编著. —北京：
清华大学出版社，2002
（计算机专业考研系列教材）
ISBN 7-302-06056-8

I. 数… II. 李… III. 数据结构-研究生-入学考试
-教材 IV. TP311.12

中国版本图书馆 CIP 数据核字（2002）第 086604 号

出版者 ： 清华大学出版社（北京清华大学学研大厦，邮编 100084）
http://www.tup.tsinghua.edu.cn
印刷者 ： 北京市耀华印刷有限公司
发行者 ： 新华书店总店北京发行所
开 本 ： 787×1092 1/16 印张：20.875 字数：508 千字
版 次 ： 2003 年 1 月第 1 版 2003 年 1 月第 1 次印刷
书 号 ： ISBN 7-302-06056-8/TP・3611
印 数 ： 0001~6000
定 价 ： 28.00 元

《计算机专业考研系列教材》丛书序

计算机专业是当今最热门也是发展最迅速的学科之一，很多学生为了进一步提高专业水平和应用能力，纷纷报考计算机专业研究生。据统计，近几年报考计算机软件与理论、计算机应用和计算机与通信专业硕士研究生的考生远远超过报考其他专业的考生，其中有相当一部分考生原来所学并非计算机专业，还有很多考生是工作多年的在职人员。为了方便报考者复习计算机专业课程，我们特地组织一批计算机专业教学第一线的教授和副教授（其中大多数编写者多年参与硕士研究生入学专业试卷命题工作）编写了本套丛书。丛书包括：

1. 《C程序设计考研指导》
2. 《数据结构考研指导》
3. 《操作系统考研指导》
4. 《编译原理考研指导》
5. 《计算机组成原理考研指导》
6. 《离散数学考研指导》

本丛书具有以下特点：

➤ 讲述全面而详实

本丛书全面涵盖各门专业课程的内容。不针对个别学校的命题特点，而是充分地讲授课程中的重点、难点和考点，并通过例题进行扩充与深化，使读者得以全面温习与提高，不留“死角”。

➤ 阐述简洁明了

不同于本、专科教材，本丛书旨在使考生花较少的时间温习各门课程的内容，因此，不过多地解释简单的术语，尽可能地高度概括和总结基本概念，使读者将主要精力集中在解题过程中。

➤ 重点突出解题思路

本丛书重点介绍解题的方式方法，不仅授人以“鱼”，更授人以“渔”。书中所选的例题和习题大多是计算机专业研究生入学考试试题（题目前标有“▲”号），并配上详解，具有很强的实战性。

➤ 强调内容的综合与提高

一般的教科书，按照内容的先后顺序按部就班地介绍。这种方式有助于初学者学习，但不便于综合复习，而考研题一般都具有很强的综合性，往往一道题涉及一本书中的好几个概念。本丛书打破了普通教材这种局限性，将相关的概念有机地融于一体，从而提高考生的解题能力。

➤ 答疑解惑

本丛书所选择的例题和习题大部分都具有较高的难度，书中不仅给出了答案，而且详细介绍解题思路和解题过程，有助于澄清概念和纠正误区。

尽管目前已有一些考研类参考书，但专门针对考研的系列教材还很少，本丛书希望在这方面作一些探索和尝试，起到抛砖引玉的作用。敬请广大读者和同行不吝赐教。

丛书编委会
2002年10月

前　言

随着全国考研课程的改革，研究生入学考试由原来的5门课程改为政治、外语、基础课和专业基础课等4门课程，其中不再包含专业课（改为在复试中考试专业课）。作为计算机专业的一门重要的专业基础课程——“数据结构”，在计算机专业研究生入学考试中占有重要地位，成为绝大多数高校招收计算机及相关专业硕士研究生的必考科目之一。

本书是作者针对数据结构课程内容丰富、学习量大、其原理和算法十分抽象的特点，在总结长期教学经验、分析考研试题的基础上编写的。全书分为11章，第1章为绪论，介绍数据结构的基本概念，特别强调算法分析的方法；第2章为线性表，介绍线性表的两种存储结构即顺序表和链表的逻辑结构与基本运算的实现过程；第3章为栈和队列，介绍这两种特殊的线性结构的概念与应用；第4章为串，介绍串的概念与模式匹配算法；第5章为数组和广义表，介绍多维数组、稀疏矩阵和广义表的概念与基本运算的实现过程；第6章为树和二叉树，介绍树和二叉树的概念与各种运算的实现过程，这一章是考研的重点，特别强调递归算法的实现过程；第7章为图，介绍图的概念与图的各种运算的实现过程；第8章为查找，介绍各种查找算法的实现过程；第9章为内排序，介绍各种内排序算法的实现过程；第10章为外排序，介绍各种外排序算法的实现过程；第11章为文件，介绍各类文件的组织结构。每章的开头给出了该章的核心考点和考试频度，其中5个“★”号表示考试频度最大。

除了介绍数据结构的基本内容外,各章中还给出一些“基础要点”，列出了本章或本节中包含的一些最重要的概念，这些概念往往在考题中以选择题或填空题的形式出现。书中还收集了近几年来大量的高校数据结构考研试题（这些题目前加有“▲”号），并给出了详解或参考答案。最后在附录A中给出一份全真《数据结构》研究生入学考试试题及参考答案。

尽管本书是专为考研的读者编写的，但也适合于作为计算机及相关专业本科生“数据结构”课程的教材。

由于作者水平所限，本书不免存在疏漏和不足之处，敬请读者批评指正。

作　者

2002年9月

目　录

第1章　绪　论

核心考点：数据结构的定义；算法的特性和分析算法的时间复杂度。
考试频度：★★

1.1 什么是数据结构

数据是指能够被计算机识别、存储和加工处理的信息的载体。

数据元素是数据的基本单位，有时一个数据元素可以由若干个数据项组成。数据项是具有独立含义的最小数据标识单位。例如，在整数这个集合中，10就是一个数据元素。又比如在一个数据库(关系数据库)中，一个记录可称为一个数据元素，而这个元素中的某一字段就是一个数据项。

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。这些数据元素不是孤立存在的，而是有着某种关系，这种关系称为结构。根据数据元素之间关系的不同，可分为下列四类基本结构：

（1）集合：结构中的数据元素同属于一个集合(集合类型由于元素之间的关系过于松散，而在数据结构课程中较少讨论)。
（2）线性结构：结构中的数据元素存在一个对一个的关系。
（3）树形结构：结构中的数据元素存在一个对多个的关系。
（4）图形结构：结构中的数据元素存在多个对多个的关系。

归纳起来，数据结构包括以下三方面内容：逻辑结构、存储结构和数据运算。

1.1.1 逻辑结构

逻辑结构是对数据之间关系的描述，所以有时就把数据的逻辑结构简称为数据结构。逻辑结构在形式上用二元组：

```
B=(K, R)
```

来表示，其中K是结点(数据元素)的有穷集合，即K是由有限个结点构成的集合；R是K上关系的有穷集合，即R是由有限个关系构成的集合，而每个关系都是从K到K的关系。在表示关系时，用尖括号表示有向关系，如<a，b>表示存在结点a到结点b之间的关系；用圆括号表示无向关系，如(a，b)表示既存在结点a到结点b之间的关系，又存在结点b到结点a之间的关系。设r是一个K到K的关系，r∈R，若k，k'∈K，且<k，k'>∈r，则称k'是k的后继结点，k是k'的前驱结点，这时k和k'是相邻的结点(都是相对r而言的)；如果不存在一个k'使<k，k'>∈r，则称k为r的终端结点；如果不存在一个k'使<k'，k>∈r，则称k为r的开始结点；如果k既不是终端结点也不是开始结点，则称k是内部结点。

例如，我们称数据库中的一个表为一个数据结构，它由很多记录(数据元素)组成，每个元素又包括多个字段(数据项)。那么这张表的逻辑结构是怎么样的呢？我们分析数据结构都是从结点(其实也就是元素、记录、顶点，虽然在各种情况下所用名字不同，但说的是同一个概念)之间的关系来分析。对于这个表中的任一个记录(结点)，它只有一个前驱结点，也只有一个后继结点，整个表只有一个开始结点和一个终端结点，当我们知道了这些关系之后，就能明白这个表的逻辑结构，即为线性结构了。

【例1.1】 设数据逻辑结构如下：

```
B1=(K,R)
K={1,2,3,4,5,6,7,8,9}
R={r}
r={<1,2>,<1,3>,<3,4>,<3,5>,<4,6> ,<4,7>,<5,8>,<7,9>}
```

试画出对应的逻辑结构图，并给出哪些是开始结点，哪些是终端结点，说明是何种数据结构。

【解】 B1对应的逻辑结构图如图1.1所示。其中，1是开始结点；2，6，8，9是终端结点。它是一种树形结点。

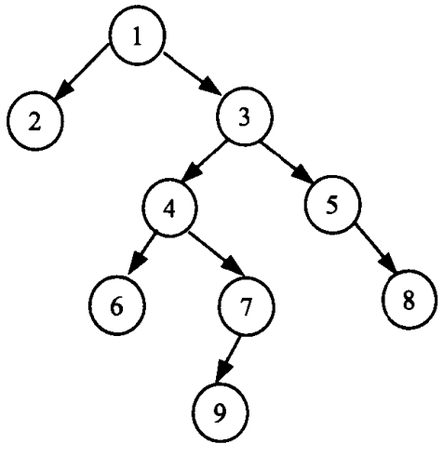

图 1.1 逻辑结构图

1.1.2 存储结构

存储结构是指如何用计算机语言表示结点之间的各种关系。对于前面的表，在计算机语言中可以描述为连续存放在一片内存单元中，也可以随机地存放在内存中再用指针把它

们连接在一起。这两种表示方法构成两种不同的存储结构。

数据的存储方式有四种：顺序存储方式、链式存储方式、索引存储方式和散列存储方式。

顺序存储方式是把逻辑上相邻的结点存储在物理上相邻的存储单元里，结点之间的关系由存储单元的邻接关系来体现。其优点是占用最少的存储空间；其缺点是由于只能使用相邻的一整块存储单元，因此可能产生较多的碎片现象。

链式存储方式是将结点所占的存储单元分为两部分，一部分存放结点本身的信息，即为数据项；另一部分存放该结点的后继结点所对应的存储单元的地址，即为指针项。其优点是不会出现碎片现象，充分利用所有的存储单元；其缺点是每个结点占用较多的存储空间。

索引存储方式是用结点的索引号来确定结点存储地址。其优点是检索速度快。其缺点是增加了附加的索引表，会占用较多的存储空间；另外，在增加和删除数据时由于要修改索引表因而会花费较多时间。

散列存储方式是根据结点的值确定它的存储地址。其优点是检索、增加和删除结点的操作都很快；其缺点是采用不好的散列函数时可能出现结点存储单元的冲突，为解决冲突需要附加的时间和空间开销。

【例1.2】 以下关于数据的存储结构的叙述中，哪些是正确的？(本题是计算机软件水平考试试题，有改动)

① 顺序存储方式只能用于存储线性结构。
② 顺序存储方式的优点是存储密度大，且插入、删除运算效率高。
③ 链表的每个结点中都恰好包含一个指针。
④ 散列法存储的基本思想是由关键字的值决定数据的存储地址。
⑤ 散列表的结点只包含数据元素自身的信息，不包含任何指针。
⑥ 栈和队列的存储方式既可是顺序方式，也可是链式方式。

【解】 ① 错误(如完全二叉树可以采用顺序存储方式存储)。
② 错误(顺序存储由于是一组连续的存储单元按顺序存储，插入和删除需大量移动记录，执行效率低)。
③ 错误(如双链表有两个指针)。
④ 正确(散列法就是用散列函数作用于关键字值产生数据存储地址)。
⑤ 错误(散列表在处理“冲突”时，可用拉链法，这样需使用一个指针)。
⑥ 正确(栈和队列都是逻辑定义，只要能满足先进后出和先进先出关系的存储方式，如顺序方式、链式方式都可采用)。

1.1.3 数据运算

数据运算就是施加于数据的操作。比如，有一张表格，我们需要对它进行查找、增加、修改、删除记录等工作，而怎么样才能进行这样的操作呢?

这就需要进行数据运算。在数据结构中，运算不仅仅是加减乘除这些算术运算，还常常涉及算法问题。算法的实现是与数据的存储结构密切相关的。

数据结构概念的基础要点

（1）数据结构被形式地定义为(K，R)，其中K是数据元素的有限集合，R是K上关系的有限集合。

（2）▲数据结构包含三个方面的内容，即数据的逻辑结构、存储结构(或物理结构)和对数据进行的运算或操作。

（3）▲数据结构中有4种基本结构，它们是集合、线性结构、树形结构和图形结构。

（4）数据在计算机内的表示称为数据的物理结构(或存储结构)。

（5）▲在数据的物理结构中，结点可以有四种存储方式，分别是顺序存储方式、链式存储方式、索引存储方式和散列存储方式。

（6）在线性结构中，第一个结点没有前驱结点，其余每个结点有且只有一个前驱结点；最后一个结点(终端结点)没有后继结点，其余每个结点有且只有一个后继结点。

（7）▲在树形结构中，树根结点没有前驱结点，其余每个结点有且只有一个前驱结点；叶子结点没有后继结点，其余每个结点的后继结点可以是任意多个。

（8）在图形结构中，每个结点的前驱结点数和后继结点数可以是任意多个。

（9）▲线性结构中元素之间存在一对一关系，树形结构中存在一对多关系，图形结构中存在多对多关系。

（10）▲在循环队列、链表、哈希表和栈中，与数据的存储结构无关的是哈希表。

1.2 算法和算法分析

1.2.1 算法及其表示

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，其中每条指令表示一个或多个操作。算法有以下五个重要特征：

（1）有穷性：一个算法必须总是(对任何合法的输入值)在执行有穷步之后结束，且每一步都可在有穷时间内完成。

（2）确定性：算法中每一条指令必须有确切的含义，不会产生二义性。

（3）可行性：一个算法是能行的，即算法中描述的操作都是可以通过已经实现的基本运算的有限次执行来实现。

（4）输入性：一个算法有零个或多个的输入。

（5）输出性：一个算法有一个或多个的输出。

【例1.3】 ▲考虑下列两段描述：

```
(1) void exam1()
    {
       n=2;
       while (n%2==0)
            n=n+2;
```

```
(2) void exam2()
       {
          y=0;
          x=5/y;
          printf("%d,%d\n",x,y);
```

```
        printf("%d\n",n);              }
    }
```

这两段描述均不能满足算法的特征，试问它们违反了哪些特征？

【解】 （1）是一个死循环，违反了算法的有穷性特征。

（2）包含除零错误，违反了算法的可行性特征。

描述算法的方法很多，有的采用类PASCAL，有的采用自然语言等。本书采用C语言来描述算法的实现过程。

【例1.4】 试编写一个算法，从大到小依次输出顺序读入的三个整数x，y和z的值。

【解】 依次输入x，y和z这三个整数，通过比较交换后，使得x≥y≥z，然后输出x，y，z。在算法中应考虑对这三个元素作尽可能少的比较和移动，如下列算法在最坏的情况下只需进行3次比较和7次移动。

```
void Descending()
{
    printf("输入x,y,z");
    scanf("%d,%d,%d",&x,&y,&z);
    if (x<y) {
        temp=x;x=y;y=temp;        //交换x，y，使x>=y
        }
    if (y<z) {
        temp=z;
        z=y;
        if (x>=temp) y=temp;
        else {
            y=x;x=temp;
        }
    }
    printf("%d,%d,%d\n",x,y,z);
}
```

1.2.2 算法分析

算法分析的两个主要方面是算法的时间复杂度和空间复杂度，其目的主要是考察算法的时间和空间效率，以求改进算法或对不同的算法进行比较。一般情况下，鉴于运算空间(内存)较为充足，所以把算法的时间复杂度作为分析的重点。

所谓一个语句的频度，即指该语句在算法中被重复执行的次数。算法中所有语句的频度之和记做T(n)，它是该算法所求解问题规模n的函数。当问题的规模n趋向无穷大时，T(n)的数量级称为渐近时间复杂度，简称为时间复杂度，记作T(n)=O(f(n))。

上述表达式中“O”的含义是T(n)的数量级，其严格的数学定义是：若T(n)和f(n)是定义在正整数集合上的两个函数，则存在正的常数C和n_0，使得当$n \geq n_0$时，总是满足$0 \leq T(n) \leq C \cdot f(n)$。但是我们总是考虑在最坏情况下的时间复杂度，以保证算法的运行时间不会比它更长。

【例1.5】 ▲给出下列程序段中带标号的语句(①～⑤)执行的频度，利用大“O”记

号将以下程序段在最坏情况下的运行时间表示为n的函数：

```
for (i=1;i<=n;i++)                                    ①
    for (j=1;j<=n;j++)                                ②
      {
       c[i][j]=0;                                     ③
       for (k=1;k<=n;k++)                             ④
          c[i][j]=c[i][j]+a[i][k]*b[k][j];            ⑤
       }
```

【解】 语句①执行频度为n+1；语句②执行频度为n(n+1)；语句③执行频度为n^2；语句④执行频度为$n^2(n+1)$；语句⑤执行频度为n^3。

其算法耗费的时间是该算法中每条语句频度之和，故：

$T(n)=2n^3+3n^2+2n+1=O(n^3)$

【例1.6】 ▲给出以下算法的时间复杂度。

```
void sort(int j,int n,int a[])
{
   if (j==n)    //子数组只有一个元素                    ①
      printf("%d ",a[n]);
    else {
       for (i=j+1;i<=n;i++)                           ②
          if (a[i]<a[j]) {
             temp=a[i];a[i]=a[j];a[j]=temp;
          }
       printf("%d ",a[j]);                            ③
       sort(j+1,n,a);                                 ④
    }
}
```

【解】 从上述算法中看出：当j=n时，子数组只有一个元素，此时只需打印一个数a[n]，时间为常量，用O(1)表示；不然则执行else子句，其中语句①需要执行n-j次，语句③是打印一个数，时间为常量，用O(1)表示，语句④是递归调用语句，用T(j,n)表示sort(j,n)的工作时间，语句④的工作时间就是T(j+1,n)，由于a[j+1]~a[n]的排序时间与a[j]~a[n-1]的排序时间相等，所以

$$T(j+1,n)=T(j,n-1)$$

则有

$$T(j,n)=T(j,n-1)+n-j+1$$

其中，1为语句③的工作时间，n-j为语句②的工作时间。若用T(n)表示sort(1,n)的工作时间，则得到如下递归表达式：

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