高 职 高 专 计 算 机 系 列 教 材

数据结构

包振宇 孙干 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高职高专计算机系列教材

数据结构

包振宇　孙 干　编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本教材主要面向高职高专院校计算机类专业的学生。教材内容以“实践应用”为主体，理论以“够用”为尺度，理论与实验相结合。本教材有如下特点：

（1）所有例题都分为：示图、分析、流程图和函数四个部分组成，思路清晰，层次鲜明，能逐步培养和提高学生分析问题和解决问题的能力。

（2）以全国计算机程序员考试大纲为基准，涉及到考试的章节选用部分历年试题举例，加强学生对所学内容进一步理解、巩固和应用。

（3）本书的算法用标准 C 语言的函数来实现，加上主函数后可直接在 Turbo C 或 Visual C++ 6.0 环境下运行。

本书每章配有适量习题和实验，具有很强的针对性和可操作性。实验程序可直接在 Turbo C 或 Visual C++ 6.0 环境下运行。

图书在版编目（CIP）数据

数据结构/包振宇，孙干编著. —北京：中国铁道出版社，2005.7（2006.8 重印）

（高职高专计算机系列教材）

ISBN 7-113-06595-3

Ⅰ.数… Ⅱ.①包… ②孙… Ⅲ.数据结构-高等学校：技术学校-教材 Ⅳ.TP311.12

中国版本图书馆 CIP 数据核字(2005)第 090563 号

书　　名：数据结构

作　　者：包振宇　孙　干

出版发行：中国铁道出版社(100054，北京市宣武区右安门西街 8 号)

策划编辑：严晓舟　张　梅

责任编辑：苏　茜　翟玉峰

特邀编辑：李晓霞

封面制作：白　雪

印　　刷：北京市彩桥印刷有限责任公司

开　　本：787×1092　1/16　印张：10.75　字数：255 千

版　　本：2005 年 8 月第 1 版　　2006 年 8 月第 2 次印刷

印　　数：5 001～7 000 册

书　　号：ISBN 7-113-06595-3/TP・1548

定　　价：16.00 元

高职高专计算机系列教材

编　委　会

前　言

“数据结构”是计算机程序设计的重要基础，也是计算机相关专业的一门重要基础课程和核心课程。它对程序设计思想的建立、提升有着重要的作用，既为后续的计算机课程奠定了一个较为扎实的基础，又可提高学生分析问题和解决问题的能力。

本教材主要面向高职高专院校计算机类专业的学生。教材内容以“实践应用”为主体，理论以“够用”为尺度。全书共分理论知识和实验指导两个部分，理论知识部分又分为 8 章：第 1 章绪论；第 2 章线性表；第 3 章栈和队列；第 4 章字符串、数组和广义表；第 5 章树；第 6 章图；第 7 章查找；第 8 章排序。实验部分精选 15 个典型实例帮助学生巩固所学知识。

本教材有如下特点：

（1）所有例题都分为：示图、分析、流程图和函数 4 个部分组成，思路清晰，层次鲜明，能逐步培养和提高学生分析问题和解决问题的能力。

（2）以全国计算机程序员考试大纲为基准，涉及到考试的章节选用部分历年试题举例，加强学生对所学内容进一步理解、巩固和应用。

（3）本书的算法用标准 C 语言的函数来实现，加上主函数后可直接在 Turbo C 或 Visual C++ 6.0 环境下运行。

（4）本书每章配有适量习题和实验，具有很强的针对性和可操作性。实验程序可直接在 Turbo C 或 Visual C++ 6.0 环境下运行。

建议本教材理论教学时数为 40 学时左右，实验教学时数为 30 学时左右。

本教材理论部分的第 1、2、3、4、5、8 章和实验部分由包振宇编写，理论部分的第 6、7 章和习题部分由孙干老师编写，最后由包振宇完成全书统稿。

由于时间仓促和编者水平有限，书中难免存在不妥或疏漏之处，敬请广大读者与专家不吝赐教。

编　者

2005 年 7 月

目　　录

第一部分　理论知识

第二部分　实验部分

第一部分　理论知识

第 1 章　绪论

“数据结构”形成和发展的背景：自 1946 年第一台计算机问世以来，计算机已深入到人类社会的各个领域，计算机的应用已不再局限于科学计算，而更多地用于控制管理及数据处理等非数值计算的处理工作。计算机加工处理的对象由纯粹的数值发展到字符、表格和图像、声音等各种具有一定结构的数据，这就给程序设计带来一些新的问题。为编写出一个“好”的程序，必须分析待处理的对象的特性以及各处理对象之间存在的关系，这就是“数据结构”这门学科形成和发展的背景。

1.1　数据结构与算法

1.1.1　数据结构的基本概念

1. 数据

对现实世界的事物采用计算机能识别、存储和处理的形式所进行的描述。简言之，数据是计算机程序能加工和处理的对象或数据，也就是计算机化的信息。

2. 数据元素

数据的基本单位，又称为结点。在计算机程序中通常作为一个整体进行考虑和处理。有时一个数据元素可由若干个数据项组成。如一本书的书目信息为一数据元素，而书目信息中的每一项（如书名、作者名等）为一个数据项。数据项是数据不可分割的最小单元。

3. 数据对象

性质相同的数据元素的集合是数据的一个子集。

4. 数据结构

（1）相互之间存在一种或多种特定关系的数据元素的集合。数据结构是指数据对象及其相互关系和构造方法。一个数据结构 DS 可以被形象地用一个二元组表示为：DS=(D,R)，其中：D 是数据结构中的数据（称为结点）的非空集，R 是定义在 D 上的关系的非空有限集合。结构是指结点之间的关系，数据结构就是结点的有限集合和关系的有限集合。

（2）数据结构中，结点及结点间的相互关系是数据的逻辑结构，数据结构在计算机中的存储内容，一般包括结点的值和结点间的关系，数据结构的存储形式是数据的存储结构（或物理结构）。

（3）数据结构按逻辑关系的不同分为线性结构和非线性结构两大类，其中非线性结构又分为树形结构和图结构，而树形结构又可分为树结构和二叉树结构。

（4）在计算机中表示信息的最小单位是二进制数的一位叫做位（bit），可以用一个或若干位组合起来形成的一个位串表示一个数据元素或结点。元素或结点可看成是数据元素

在计算机中的映象有顺序映象、非顺序映象，相应地，存储结构可分为顺序存储结构、链式存储结构。顺序映象的特点是借助元素在存储器中的相对位置来表示数据元素之间的逻辑关系。非顺序映象的特点是借助指示元素存储地址的指针表示数据元素之间的逻辑关系。

5. 数据类型

指某种程序设计语言所允许使用的变量种类。

1.1.2 算法的概念和特性

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，其中每一条指令表示一个或多个操作。在计算机科学中算法则是描述计算机解决给定问题的操作过程。

算法的特性是有穷性、确定性、可行性、输入和输出。

（1）有穷性：求解问题的步骤是有限的，且每一步都可在有限时间内完成。

（2）确定性：在任何情况下，对同一问题的求解结果必须是唯一的。

（3）可行性：算法的实现必须在法律上、经济上、技术上都可行。

（4）输入：没有输入，对问题的求解方法仅能适用于这一个特定问题。

（5）输出：没有输出，看不到结果，对问题的求解将失去任何意义。

1.2 算法的描述和分析

1.2.1 算法的描述

算法需要用一种语言来描述，同时，算法可有各种描述方法以满足不同的需求。例如，一个需在计算机上运行的程序（程序也是算法）必须严格按照语法规定用机器语言、汇编语言或高级语言编写，而一个为了供人阅读和交流的算法可以用伪码语言或框图等其他形式来描述。伪码语言是一种包括高级程序语言的 3 种基本控制结构（顺序、判定和循环）和自然语言成分的“面向对象”的语言。

本书采用 C 语言描述数据结构及相应的算法。

1.2.2 算法的分析

1. 从时间方面分析

算法执行时间需通过依据该算法编制的程序在计算机上运行时所消耗的时间来度量。一个算法是由控制结构（顺序、分支和循环 3 种）和原操作（指固有数据类型的操作）构成的，因此算法时间取决于两者的综合效果。为了便于比较同一问题的不同算法，通常的做法是，从算法中选取一种对于所研究的的问题（或算法类型）来说是基本操作的原操作，以该基本操作重复的次数作为算法的时间复杂度。算法的时间复杂度通常记作：

$T(n)=O(f(n))$

其中，n 为问题的规模，f(n)表示算法中基本操作重复执行的次数，是问题规模 n 的某个函数。上式表示随问题规模 n 的增大算法执行时的增长率与 f(n)的增长率相同。f(n)和 T(n)是同数量级的函数，大写字母 O 表示 f(n)与 T(n)只相差一个常数倍。

算法的时间复杂度用数量级的形式表示后，一般可简化为分析循环体内基本操作的执

行次数即可。

【例 1-1】（1）x++;

（2）for(i=1;i<n;i++)x++;

（3）for(i=1;i<n;i++)

for(j=1;j<n;j++)

x=x+1;

上面 3 个程序段含基本操作“x 增 1”的语句 x++的频度分别为 1，n 和 n^2 则它们的时间复杂度分别为 O(1)，O(n)和 $O(n^2)$分别称为常量阶，线性阶和平方阶。有的算法的时间复杂度有：对数阶 $O(\log 2^n)$，指数阶 $O(2^n)$等。

【例 1-2】

（1）for(i=1;i<n;i++)

for(j=1;j<n;j++)

for(k=1;k<n;k++)

x++;

程序中 x++的时间复杂度为 $O(n^3)$

（2）for(i=1;i<n;i++)

for(j=1;j<i;j++)

for(k=1;k<j;k++)

x++;

程序段中 x++的时间复杂度为 $O(\sum_{i=1}^{n}\frac{i(i+1)}{2})$

2. 从空间方面分析

类似于算法的时间复杂度，以空间复杂度作为算法所需存储空间的量度，记为：

S(n)=O(f(n))

其中，n 为问题的规模（或大小）。算法在运行过程中需辅助存储空间的大小。在这里我们不多作讨论。

习题 1

一、选择题

1. 数据的基本单位是_____。

A．数据结构　　B．文件　　C．数据元素　　D．数据项

2. 以下哪种数据结构中任何两个结点之间都没有逻辑关系_____。

A．树形结构　　B．集合　　C．图状结构　　D．线性结构

3. 要求同一逻辑结构的所有数据元素具有相同的特性，这意味着_____。

A．数据元素具有同一的特点

B．不仅数据元素包含的数据项的个数相同，而且对应数据的类型要一致

C．每个数据元素都一样

D．数据元素所包含的数据项的个数要相等

4．数据结构被形象地定义为（D,R），其中D是（1）的有限集合，R是D上（2）的有限集合。

（1）A．算法　　B．数据元素　　C．数据操作　　D．逻辑结构

（2）A．操作　　B．映像　　C．存储　　D．关系

5．在数据结构中，从逻辑上可以把数据结构分为_____。

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

6．线性表的线性结构是一种_____的存储结构。

A．随机存取　　B．顺序存取　　C．索引存取　　D．HASH存取

7．计算机算法指的是（1），它必须具备输入、输出和（2）等5个特征。

（1）A．计算方法　　B．排序方法

C．解决某一问题的有限运算序列　　D．调度方法

（2）A．可行性、可移植性和扩充性　　B．可行性、确定性和有穷性

C．确定性、有穷性和稳定性能　　D．易读性、稳定性和安全性

8．线性表若采用链表存储结构，要求内存中可用存储单元的地址_____。

A．必须是连续的　　B．部分必须是连续的

C．一定是不连续的　　D．连续不连续都可以

9．以下叙述中，正确的是_____。

A．线性表的线性存储结构优于链式存储结构

B．二维表组是它的每个数据元素为一个线性表的线性表

C．栈的操作方式是先进先出

D．队列的操作方式是先进后出

10．以下说法正确的是_____。

A．数据元素是数据的最小单位

B．数据项是数据的基本单位

C．数据结构是带有结构的各数据项的集合

D．数据结构是带有结构的数据元素的集合

二、填空题

1．所谓数据的逻辑结构指的是数据元素之间的_____。

2．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，它包含3方面内容_____。

3．数据的逻辑结构包括_____、_____、_____和_____4种类型。

4．算法的5个重要特性是_____、_____、_____、_____、_____。

5．下列程序段的时间复杂度是_____。

```
for(i=1;i<=n;i++)
   A[i]=0;
```

6．下列程序段的时间复杂度是_____。

```
s=0;
for(i=1;i<=n;i++)
  for(j=1;j<=n;j++)
   s=s+b[i][j];
sum=s;
```

7. 存储结构是逻辑结构的_____实现。

9. 常见时间复杂度的量级有：常数阶 O_____，对数阶 O_____，线性阶 O_____，平方阶 O_____和指数阶 O_____。通常认为，具有指数阶量级的算法是_____的。

9. 数据对象是性质相同的_____集合。

三、应用题

1. 分析下列程序段的时间复杂度。

```
……
i=1;
While(i<=n)
i=i*2;
……
```

2. 叙述算法的定义及其重要特性。

3. 叙述下列术语：数据，数据元素，数据结构，数据对象。

4. 逻辑结构与存储结构是什么关系？

5. 设 n 为正数。试确定下列程序段中前面加记号@的语句的频度：

```
i=1;k=0;
While(i<=n-1)
{@k+=10*i;
 i++;}
 k=0;
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
@ k++;
```

四、算法设计题

1. 已知输入 x、y、z 三个不相等的整数，设计一个算法，使得这 3 个数按从大到小的顺序输出，并考虑所用算法的比较次数和元素移动次数。

2. 编写算法实现：在输入确定的 10 个数中找出最小的数和最大的数。

第 2 章　线性表

线性表的特点是：在数据元素的非空有限集中，（1）存在唯一的一个被称为“第一个”的数据元素；（2）存在唯一的一个被称为“最后一个”的数据元素；（3）除第一个以外，集合中的每一个数据元素均有且只有一个前驱；（4）除最后一个以外，集合中每一个数据元素均有且只有一个后继。

2.1　线性表的逻辑结构

线性表是最简单的一种数据结构。简单地说，一个线性表是 n（n≥0）个具有相同属性的数据元素的有限序列，其中各个数据元素有着依次相邻（串接）的逻辑关系。

线性表中数据元素的个数 n 称为线性表的长度。n=0 时，该线性表为空表。n>0 时该线性表可记为：

（A_0,A_1,…,A_i, …,A_{n-1}）

其中，A_0 称为起点，A_{n-1} 称为终点，每个元素的序号代表它在该线性表中的逻辑位置减 1（与数组下标对应），除了起点（A0）和终点(A_{n-1})外，每个数据元素都有且只有一个直接前驱和一个直接后继。A_{i-1} 是 A_i 的直接前驱，A_{i+1} 是 A_i 的直接后继。

线性表中的数据元素可以是各式各样的，但同一线性表中的元素必定有相同的特性，即属同类数据对象，相邻数据元素之间存在着有序关系。

线性表是一个相当灵活的数据结构，其表长可根据不同的操作增长或缩短。对线性表进行的基本操作有：存取、插入、删除、合并、分解、查找、排序、求线性表的长度等。

2.2　线性表的顺序存储结构

2.2.1　顺序分配

线性表的顺序存储结构是用一组地址连续的存储单元依次存储该线性表中的各个元素。假设线性表的每个数据元素占用 L 个存储单元，并以所占的第一个单元的存储地址为数据元素的存储位置，则第 i+1 个数据元素的存储位置为：

$Loc(A_i)=Loc(A_0)+i*L$

因此，在内存中可以通过地址计算直接存取线性表中的任一元素，所以，线性表的顺序存储结构可以随机存取。这种结构的特点是逻辑上相邻的元素物理上也相邻（如图 2-1 所示）。顺序表可用 C 语言的一维数组实现，数组的类型随数据元素的属性而定。

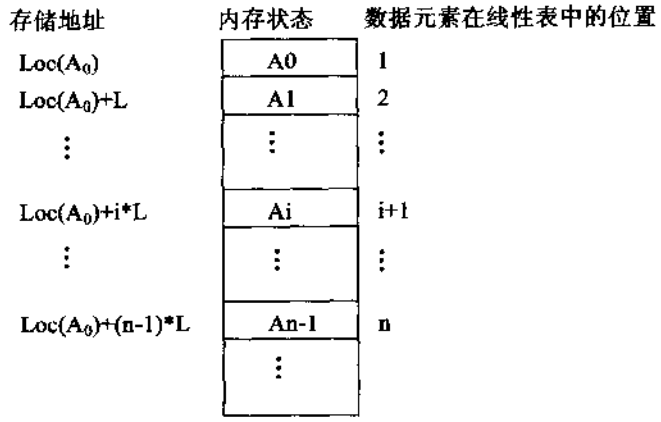

图 2-1　线性表的顺序存储结构示意图

2.2.2　线性表的操作

线性表结构简单、易于实现，但插入或删除一个元素时需对其后继的全部元素逐个进行移动（平均需移动表中的一半元素）操作不方便，需花费较多时间，特别是当插入元素而需动态扩大连续存储时，线性表后面的存储区可能已被占用从而无法扩大。所以，这种结构仅适用于数据元素个数不经常变动或只需在顺序存取设备上做成批处理的场合。下面我们分情况讨论线性表的插入和删除操作。

1. 线性表的插入操作

（1）在数组中下标为 i 的元素前插入一个新元素

【例 2-1】某 C 语言程序中，整型数组 V 的 99 个元素 V[0]～V[98]组成一个线性表。为了在 V[l]位置前插入一个新元素 b，可用函数 inst1 来实现，当插入成功时返回 1，否则返回 0，所以该函数的返回值类型是整型。插入前后的线性表的示意图如图 2-2 和图 2-3 所示。

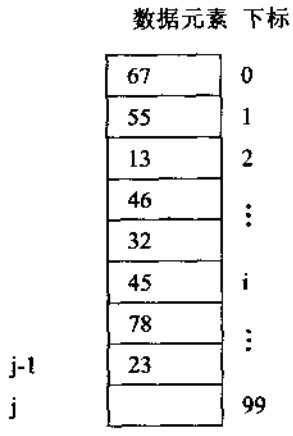

图 2-2　插入前的数组

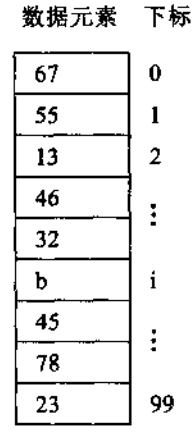

图 2-3　插入后的数组

分析：① 初始条件：V，I，b

② 执行条件：0≤i≤98

③ 执行结果：1 成功

0 不成功

④ N-S 流程图如图 2-4 所示。

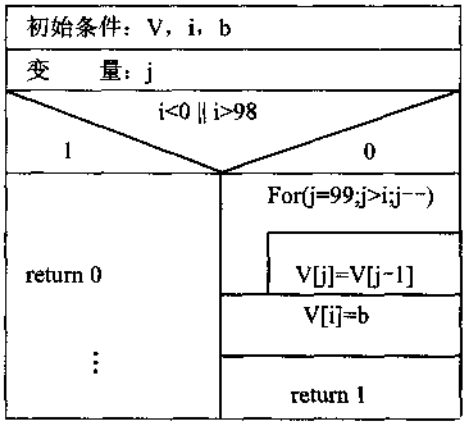

图 2-4　算法对应的 N-S 流程图

根据分析用函数实现算法如下：

```
int ins1(int V[ ], int i, int b)
{ int j;
   if(i<0||i>98)
   { printf( "The value of i is out of range\n" );
     return 0;                    /*插入失败*/
   }
  for(j=99;j>i;j--)
  v[j]=v[j-1];                    /*后移*/
  v[i]=b;                         /*插入*/
  return 1;                       /*插入成功 */
}
```

（2）在有序表中插入一个新元素

【例 2-2】设有一个有序线性表，它用数组结构实现，最大元素个数为 n。设当前元素个数是 m。要求在该有序表中插入一个值为 x 的元素。当 x=63 时，插入前后的有序表的变化如图 2-5、图 2-6 所示。