普通高等教育“十二五”创新型规划教材

数据结构

SHUJU JIEGOU

主编 金伊 金锋

北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

普通高等教育“十二五”创新型规划教材

数据结构

主　编　金　伊　金　锋
副主编　王　冰　孙　颖　于　霞

北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

内容简介

数据结构是计算机学科的必修课程。本书是作者在总结数据结构的教学和科研成果基础上，在深入学习和研究了国内外同类教材后而编写的。全书分为10章，内容包括数据结构的基本概念、线性表、栈和队列、数组和矩阵、串、广义表、树与二叉树、图、查找和排序。书中各章后都给出了难度适中的不同类型的习题，供学生课后练习使用。

本书采用C语言作为数据结构和算法的描述语言，考虑到算法描述的简洁性和知识的延续性，在算法中适当地引进了部分C++的基本概念，使得算法描述更为简明清晰。

本书可作计算机类专业的本科教材，或作为信息类相关专业的选修教材，也可作从事计算机软件开发和应用的工程技术人员参考。

图书在版编目（CIP）数据

数据结构/金伊，金锋主编. —北京：北京理工大学出版社，2012.1
ISBN 978-7-5640-5497-7

Ⅰ.①数…　Ⅱ.①金…②金…　Ⅲ.①数据结构-高等学校-教材
Ⅳ.①TP311.12

中国版本图书馆CIP数据核字（2012）第001763号

出版发行／北京理工大学出版社
社　　址／北京市海淀区中关村南大街5号
邮　　编／100081
电　　话／(010)68914775(办公室)　68944990(批销中心)　68911084(读者服务部)
网　　址／http://www.bitpress.com.cn
经　　销／全国各地新华书店
印　　刷／北京市兆成印刷有限责任公司
开　　本／710毫米×1000毫米　1/16
印　　张／12.5
字　　数／230千字
版　　次／2012年1月第1版　2012年1月第1次印刷
印　　数／1~1500册
定　　价／30.00元

责任编辑／胡　静
王玲玲
责任校对／陈玉梅
责任印制／王美丽

前言

Preface <<< <<<

数据结构是计算机、信息类及相关专业的重要专业基础课。它在整个课程体系中处于承上启下的核心地位：一方面扩展和深化在数学、程序设计语言等课程学到的基本技术和方法；另一方面为进一步学习操作系统、编译原理、数据库等专业课知识奠定坚实理论和实践基础。本课程在教给学生数据结构设计和算法设计的同时，培养抽象思维能力、逻辑推理能力和形式化思维方法，增强分析问题、解决问题和总结问题的能力，更重要的是，培养专业兴趣、树立创新意识。本教材在内容选取上符合人才培养目标的要求及教学规律，在组织编排上体现“先理论、后应用，理论与应用相结合”的原则，并兼顾学科的广度和深度，力求适用面广。

本书是笔者在总结数据结构的教学和科研成果基础上，在深入学习和研究了国内外同类教材后而编写的。全书分为 10 章，第 1 章为“绪论”，介绍数据结构的基本概念，特别强调算法的分析方法；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和应用；第 4 章为“数组和矩阵”，介绍多维数组和稀疏矩阵的概念与相关运算的实现过程；第 5 章为“串”，介绍串的概念与模式匹配算法；第 6 章“广义表”，介绍广义表的存储结构和相关算法的实现过程；第 7 章为“树与二叉树”，介绍树和二叉树的概念与各种运算的实现过程；第 8 章为“图”，介绍图的概念和图的各种运算和相关算法的实现过程；第 9 章“查找”，介绍各种查找算法的实现过程；第 10 章为“排序”，介绍多种内部排序和外部排序算法的实现过程。

本书采用 C 语言作为数据结构和算法的描述语言，考虑到算法描述的简洁性和知识的延续性，在算法中适当地引进了部分 C++的基本概念，使得算法描述更为简明清晰。

本书编写力求概念清楚、文字流畅、算法精湛、重点突出、注重应用。其中各章都给出了难易适中的不同类型的习题，供学生课后练习使用。本书适合作为计算机类专业的本科教材，也可作为信息类相关专业的选修教材。

本书由北京理工大学自动化学院李慧芳副教授主审，对书稿作了认真的审查，提出了宝贵的修改意见。在编写过程中得到禹树春副教授的关心和指导。北京理工大学出版社为本书出版做了大量工作。在此一并表示诚挚的谢意。

由于作者水平有限，加之编写时间仓促，书中难免有错误和不足，敬请读者批评指正。

编　者

目录

Contents <<< <<<

第1章 Chapter 1 绪论

数据结构是计算机及相关专业的基础课之一，是一门十分重要的核心课程，主要学习用计算机实现数据组织和数据处理的方法。数据结构也是计算机专业后续专业课（如操作系统、数据库、编译系统和软件工程）的重要基础。

本章将介绍数据结构的基本概念和基本术语、算法及算法时间复杂度的分析方法。

1.1 数据结构的基本概念

1.1.1 基本术语

数据（Data）：是所有能被计算机识别、存储和加工处理的符号的集合。数据可以是数值数据，也可以是非数值数据。数值数据是一些整数、实数或复数；非数值数据包括字符、文字、图形、语言等。

数据元素（Data Element）：数据的基本单位，在不同的条件下，数据元素又可称为元素、结点、记录等。

数据项（Data Item）：具有独立含义的数据最小单位。

一个数据元素可以由若干个数据项所组成。

例如，在表 1–1 的学生登记表中，每个学生的信息就是一个数据元素，学生信息中的每一项就是这个数据元素中的数据项。

表 1–1 学生登记表

学号	姓名	性别	年龄	籍贯
0901	张元	男	18	吉林
0902	王宇	女	19	大连
0903	马力	男	17	北京
0904	何永	男	18	河北
⋮	⋮	⋮	⋮	⋮

数据对象（Data Object）：是性质相同的数据元素的集合，是数据的一个子集。

例如，自然数的数据对象是集合{1,2,3,…}，而由 26 个英文字母组成的数据对象则是集合{A,B,C,…,Z}。

数据处理（Data Processing）：是指对数据进行查找、插入、删除、排序、计算、输入、输出等操作过程，处理的目的是获得有用的信息。

关键字（Key）：又称关键码，是指数据元素中能起标识作用的数据项，能够起唯一标识作用的关键字称为主关键字，反之称为次关键字。例如表 1–1 中，学号和姓名为关键字，其中学号为主关键字，姓名为次关键字。

1.1.2 数据结构

数据结构（Data Structure）：是指数据元素之间存在一定关系的数据元素的集合。可见，一个数据结构有两个要素：一个是数据元素的集合；另一个是关系的集合。在形式化描述中，数据结构是一个二元组。

$$DS=(D,R)$$

式中，D 是数据元素的集合，R 是 D 上关系的集合。

按着视点的不同，数据结构分为逻辑结构和存储结构。

1. 数据的逻辑结构

数据的逻辑结构（Logical Structure）是指数据元素之间的逻辑关系。根据数据元素之间逻辑关系的不同，数据的逻辑结构分为以下 4 类。

（1）集合结构（Set Structure）

在该数据结构中，只有数据元素，它们之间除了“同属一个集合”外，别无其他的关系，即 R={}。

（2）线性结构（Linear Structure）

在该数据结构中，除第一个数据元素外，其他各元素有唯一的前驱；除最后一个数据元素外，其他各元素有唯一的后续，即数据元素之间存在一对一的关系。

（3）树结构（Tree Structure）

在该数据结构中，除了一个根数据元素（结点）外，其他各元素（结点）有唯一的前驱；所有数据元素（结点）都可以有一个或多个后继，即数据元素间存在一对多的关系。

（4）图结构（Graph Structure）

在该数据结构中，各数据元素可以有多个前驱或多个后继。即数据元素间存在多对多的关系。

集合结构是数据元素之间关系的一种松散结构，树结构和图结构一般称为非线性结构。

数据的逻辑结构通常使用逻辑结构图描述：将一个数据元素看作一个结点，元素之间的逻辑关系用结点之间的连线表示，如图 1–1 所示为 4 种基本数据结构的逻辑结构。

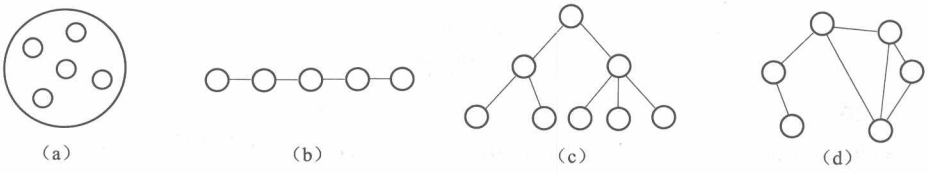

图 1–1　四种基本结构的示意图

（a）集合结构；（b）线性结构；（c）树结构；（d）图结构

数据的逻辑结构属于用户视图，是面向问题的，反映了数据内部的构成方式，与数据的存储无关，是独立于计算机的。可以看作从具体问题抽象元素的数学模型，是对操作对象的一种数学描述。

［例 1–1］有一种数据结构 B1=(D,R)，其中：

```
D= {1, 3, 5, 7, 9, 11}
R= {r}
r= {<1, 3>, <3, 5>, <5, 7>, <7, 9>, <9, 11>}
```

对应的逻辑结构图如图 1–2 所示。

图 1–2　对应 B1 的逻辑结构图

从该例可见，每个数据结点仅有一个前驱结点（除第一个结点外），仅有一个后继结点（除最后一个结点外）。这种数据结构的特点是结点之间为一对一的联系，即线性关系。这种数据结构就是线性结构。

［例 1–2］有一种数据结构 B2=(D,R)，其中：

```
D= {a, b, c, d, e, f, g, h}
R= {r}
r= {<a, b>, <a, c>, <b, d>, <b, e>, <c, f>, <c, g>, <c, h>}
```

对应的逻辑结构图如图 1–3 所示。

从该例中看出，每个结点（除根结点外）只有一个前驱结点，但有多个后继结点。这种数据结构的特点是数据元素之间为一对多关系，即层次关系。这种数据结构就是树形结构。

［例 1–3］有一种数据结构 B3=(D,R)，其中：

```
D= {a, b, c, d, e}
R= {r}
r= {(a, b), (a, c), (b, c), (c, d), (c, e), (d, e)}
```

对应的逻辑结构图如图 1–4 所示。

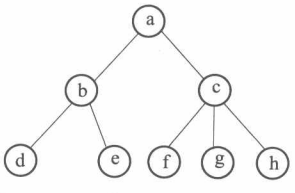

图 1–3　对应 B2 的逻辑结构图

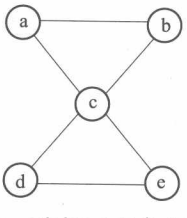

图 1–4　对应 B3 的逻辑结构图

从该例中看出，每个结点可以有多个前驱结点和多个后继结点。这种数据结构的特点是数据元素之间为多对多联系，即图形关系。这种数据结构就是图形结构。

2. 数据的存储结构

数据的存储结构（Storage Structure）又称为数据的物理结构，是指数据的逻辑结构在计算机中的存储表示，包括数据元素本身的存储表示及其逻辑关系的存储表示。数据有 4 种不同的存储结构：顺序存储结构、链式存储结构、索引存储结构和哈希存储结构。

（1）顺序存储结构

把逻辑上相邻的数据元素存储在物理位置上相邻的存储单元里，数据元素之间的逻辑关系由存储单元的邻接关系体现。顺序存储结构通常借助程序语言的数组描述。

（2）链式存储结构

把逻辑上相邻的数据元素存储在物理位置上任意的存储单元里，数据元素之间的逻辑通过附加的指针字段表示。链式存储结构通常借助于程序语言的指针类型描述。

（3）索引存储结构

在存储数据元素的同时，还建立附加的索引表。索引表中的每一项称为索引项，索引项的一般形式是：（关键字，地址），关键字唯一标识一个数据元素，地址作为指向数据元素的指针。

（4）哈希（或散列）存储结构

根据数据元素关键字直接计算出该数据元素的存储地址。把数据元素关键字 K 作为自变量，通过一个称为哈希函数 $H_{(K)}$ 的计算规则，得到的哈希函数值即为该数据元素实际存储单元地址。例如，数据元素关键字序列为（18,60,43,54,46），哈希函数 $H_{(K)}$=K MOD 13，哈希表长度为 13，通过计算可知：$H_{(18)}$=5，$H_{(60)}$=8，$H_{(43)}$=4，$H_{(54)}$=2，$H_{(46)}$=7。

上述 4 种结构即可以单独使用，也可以组合起来对数据结构进行存储映像。图 1–5 描述了数据的 4 种存储结构。

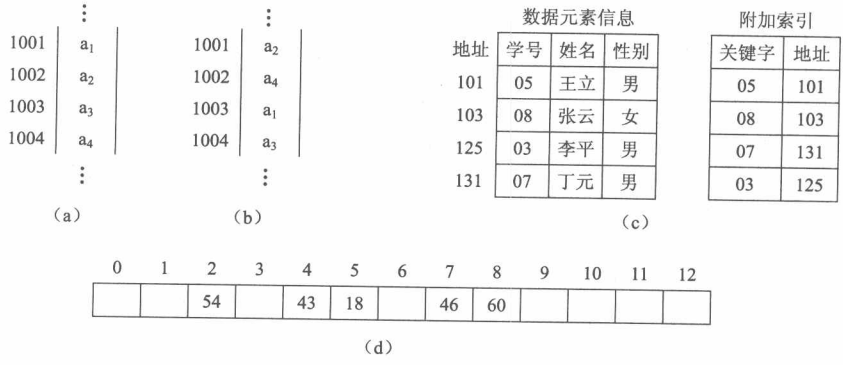

图 1–5　数据的四种存储结构

（a）顺序存储结构；（b）链式存储结构；（c）索引存储结构；（d）哈希存储结构

数据的存储结构或数据具体实现的视图，是面向计算机的，其基本目标是将数据及其逻辑关系存储到计算机的内存中，依赖于计算机语言。

1.1.3 研究数据结构的方法

研究数据结构是为了编写解决问题的程序。用计算机求解一个实际问题的过程可用图 1–6 所示的流程加以描述。

图 1–6 计算机求解问题的流程

首先从现实问题出发，抽象出一个适当的数学模型，然后设计一个求解此数学模型的算法，最后根据这个算法编出程序，经过调试、排错、运行直至得到最终的解答。问题、数学模型、算法和程序是问题求解过程中出现的 4 个不同的概念。

（1）问题

问题就是一个要完成的任务，即对应一组输入有一组对应的输出。只有在问题被准确定义并完全理解后才有可能研究问题的解决方法。

（2）数学模型

问题的数学模型是指用数学的方法精确地把问题描述成为函数。而函数是输入与输出之间的一种映射关系。函数的输入是一个值或一些信息，这些值组成的输入称为函数的参数。不同的输入可以产生不同的输出，但对于给定的输入，每次计算函数时得到的输出必须相同。

（3）算法

算法是解决某个问题的一种方法。如果将问题抽象为数学模型，那么它仅是精确地定义了输入和输出的映射关系，而算法则能把输入转化为输出。

（4）程序

一个计算机程序被认为是对一个算法用某种程序设计语言的具体实现。本书中使用以 C 语言为基本构架的程序作为算法的描述。

在问题求解模型中关键的一步是建立数学模型，而数据结构实际上就是数学模型。对算法的研究也是数据结构课程的主要内容之一。

1.2 抽象数据类型

1.2.1 数据类型

数据类型（Data Type）是一组值的集合，以及定义于这个值集上的一组操作

的总称。数据类型规定了该类型数据的取值范围和对这些数据所能采取的操作。例如，C 语言中的“整数类型”就定义了一个整数可取值的范围（其最大值 INT-MAX 依赖于具体机器），以及对整数可施加的加（+）、减（–）、乘（*）、除（/）和求模（%）等操作。

在高级程序设计语言中，数学模型可分为两类：一类是原子类型；另一类则是结构类型。原子类型的值是不可分解的。例如，C 语言中的整型、字符型、浮点型、双精度型等基本类型，分别用保留字 int、char、float、double 标识。而结构类型的值是由若干成分按某种结构组成的，因此是可分解的。例如，数组的值由若干分量组成，每个分量可以是整数，也可以是数值等。

1.2.2 抽象数据类型

抽象（Abstract）就是抽出问题本质的特征而忽略非本质的细节，是对具体事物的一个概括。例如，地图是对它所描述地域的一种抽象。

抽象数据类型（Abstract Data Type，ADT）是一个数据结构以及定义在该结构上的一组操作的总称。而数据结构又包括数据元素及元素间的关系，因此抽象数据类型一般可以由元素、关系及操作 3 个要素来定义。另外对一个抽象数据类型进行定义时，必须给出它的名字及其各操作的操作符名，即函数名，并且规定了这些函数的参数性质。一旦定义了一个抽象数据类型及具体实现，程序设计中就可以像使用基本数据类型那样，十分方便地使用抽象数据类型。因此可以说，ADT 可以理解为对数据类型的进一步抽象，数据类型和 ADT 的区别仅在于：数据类型指的是高级程序设计语言支持的基本数据类型，而 ADT 指的是自定义的数据类型。

抽象数据类型可用（D,S,P）三元组表示。其中，D 是数据对象；S 是 D 上的关系集；P 是对 D 的基本操作集。抽象数据类型的一般格式如下：

```
ADT  抽象数据类型名
{      数据对象：<数据对象的定义>
       数据关系：<数据关系的定义>
       数据操作：<数据操作的定义>
}ADT  抽象数据类型名
```

其中，数据对象和数据关系的定义用集合描述，基本操作的定义格式为：

返回类型，基本操作名（参数表）

对于每个操作，包含下列 5 个基本要素。

① 输入：对输入数据的说明。

② 前置条件：描述了操作执行之前数据结构和参数应满足的条件。若不满足，则操作失败，并返回相应出错信息。

③ 过程：对数据执行的操作。

④ 输出：对返回数据的说明。

⑤ 后置条件：执行本操作后系统的状态，“系统”可看作是某个数据结构。

基本操作有两种参数：赋值参数只为操作提供输入值；引入参数以“&”符号开头，除可提供输入值外，还将返回操作结果。

[例 1–4] 抽象数据类型 Complex 可以定义如下形式：

```
ADT Complex
{    数据对象  D={e1,e2/e1,e2∈RealSet}
     数据关系  R={<e1,e2 > | e1是复数的实数部分，e2是复数的虚数部分}
     基本操作
         Init Complex (&Z, V1, V2)
            操作结果：构造复数 Z，其实数部分和虚数部分分别被赋予参数 V1 和 V2 值
         Destroy Complex (&Z)
            操作结果：复数 Z 被销毁
         GetReal (Z, & realpart)
            初始条件：复数已存在
            操作结果：用 realpart 返回复数 Z 的实部值
         Get Imag (Z, & Imagpart)
            初始条件：复数已存在
            操作结果：用 Imagpart 返回复数 Z 的虚部值
        Add(Z1,Z2,&sum)
            初始条件：Z1 和 Z2 是复数
            操作结果：用 sum 返回复数 Z1 和 Z2 的和
 }ADT Complex
```

1.3 算　法

1.3.1 算法概述

1. 算法定义

算法（Algorithm）是为了解决某一问题而规定的一个有限长的操作序列。它是对特定问题求解步骤的一种描述，是指令的有限序列。每一条指令表示一个或多个操作。

2. 算法特性

① 有穷性。一个算法必须在执行有穷步之后结束，即必须在有限时间内完成。

② 确定性。算法的每一步必须有确切的定义，无二义性。对于相同的输入，算法在任何时间任何情况下都应该始终给出相同的输出。

③ 可行性。算法中的每一步都可以通过已经实现的基本运算的有限次执行得以实现。

④ 输入。一个算法具有零个或多个输入，这些输入取自特定的数据对象集合。

⑤ 输出。一个算法具有一个或多个输出，这些输出同输入之间存在某种特定