高等学校教材

数据结构

DATA STRUCTURE

◎ 蒋秀英　栾晓春　燕孝飞　编著

中国石油大学出版社

校教材

数据结构

DATA STRUCTURE

◎ 蒋秀英　栾晓春　燕孝飞　编著

中国石油大学出版社

内容简介

本书主要包括数据结构的基本概念、基本的数据结构（线性表、栈和队列、串、数组与广义表、树、图）以及基本技术（查找算法与排序算法）等三个部分。本书采用了面向对象的方法讲述了数据结构中的技术，并使用了读者熟悉的标准 C 作为算法描述的语言。本书取各类大型考试范围的交集，内容丰富、概念清楚、技术实用，配有大量的例题、习题、实训项目，注重考点的讲解和学生创新实践能力的培养。

本书为“山东省高等学校教学改革立项项目（2009436）”的成果之一，是在作者多年教学实践的基础上编写而成的。本书适合作为普通高等院校、高职高专院校计算机及其相关专业的教材，也可作为从事相关工作的人员学习数据结构知识的自学教材或参考书。

图书在版编目（CIP）数据

数据结构 / 蒋秀英等编著. —东营 : 中国石油大学出版社，2011.7

ISBN 978-7-5636-3547-4

Ⅰ. ①数… Ⅱ. ①蒋… 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(2011)第 164171 号

书　　名：数据结构
编　　著：蒋秀英　栾晓春　燕孝飞

责任编辑：刘　静
封面设计：赵志勇

出 版 者：中国石油大学出版社（山东 东营，邮编 257061）
网　　址：http://www.uppbook.com.cn
电子邮箱：cbs2006@163.com
印 刷 者：日照日报印务中心
发 行 者：中国石油大学出版社（电话 0546 8391810）
开　　本：185 mm × 260 mm　印张：20　字数：537 千字
版　　次：2011 年 7 月第 1 版第 1 次印刷
定　　价：36.00 元

前言 PREFACE

20世纪80年代初，“数据结构”课程就成为国内计算机专业教学计划中的核心课程。目前，ACM/IEEE CC-2004教程仍将算法与数据结构类课程列为核心课程之首。数据结构前承高级语言程序设计和离散数学，后接操作系统、编译原理、数据库原理等专业课程。数据结构是计算机专业考研、升本的必考课程，也是程序设计竞赛、“全国计算机技术与软件专业技术资格（水平）考试”的重要内容。数据结构越来越显示出其在信息学科中的重要地位。

在教学过程中，我们发现，作为各类考试都会涉及的课程，多数《数据结构》教材存在例题、习题偏少和内容与实际应用脱节等问题。本书定位于普通本科院校计算机专业教材，书中既包含了数据结构的基本概念和相关算法，又加入了大量的应用习题，重视算法的实现和课程的综合实践运用，集教材、习题和实训于一体，尽量让使用本书的学生或自学者一书在手就能够进行数据结构课程的学习和实验训练。

本书采用面向对象的方法讲述了数据结构中的技术，使用读者熟悉的标准C作为算法描述的语言，将读者的注意力集中在算法的理解上。其中，所列算法只需补充上相应的类型定义与调用，就可成为可直接在计算机上运行的C程序。教材内容取各类大型考试范围的交集，内容丰富、概念清楚、技术实用，配有大量的例题、习题、实训项目，注重考点的讲解和学生创新实践能力的培养。

本书主要包括数据结构的基本概念、基本的数据结构（线性表、栈和队列、串、数组与广义表、树、图）以及基本技术（查找算法与排序算法）等内容。在教材编写过程中，作者参阅了国内外很多经典著作和教材，并结合了多年讲授数据结构课程的经验，力求在章节安排和内容描述上突出自己的特点，使本书能真正成为易懂易学的教材。

本书主要特点：

理论知识要点突出，教材内容实用，取各类大型考试范围的交集，在保证数据结构课程体系的前提下，大胆舍去一些老版教材中的陈旧知识，使本书更加适用于普通本科院校的教学和学生自学。同时，对于一些重大考试中经常出现但很少在以往教材中出现的知识点，作者都尽所能做了详细的阐述。例如，二叉排序树和平衡二叉树的关系、哈希表查找失败的平均查找长度的求解等。

对于数据结构知识体系中的重点和常见考点，在书中以“本章教学重点和要求”和“注意”的形式体现，方便读者抓住重点，掌握相关内容，减少了自学的难度。

知识编排条理清晰、实用、易用。每章均按照“本章教学重点及要求、章节内容、小结、

习题、实训项目”体例编写，让读者开篇知晓要求，带着问题学习。小结有助于对所学内容进行归纳，通过实训和习题帮助读者训练并掌握相关的知识点。

注重实践创新能力的培养。每一章最后的实训则是一个能够涵盖整个章节上机内容的实验题目，附录中的课程设计也以实际应用为目的进行选取。结合了学科竞赛的练习要求，着力加强综合性、创新性实验，旨在提高学生的应用能力和创新水平。

每章均有大量的习题，包含了近几年较有代表意义的程序员考试和研究生考试的题目。另外，还把三年来计算机研究生统考的所有真题都按照章节进行了分解，作为习题并做了标注，方便读者检验自己对数据结构知识的掌握情况。

将“软考”认证考试内容引入了教材，满足学生的就业需求。

附上了最新的考研大纲、ACM 程序设计竞赛等的要求。

总之，本书内容丰富、实用，语言简练，注重考点和培养学生创新实践能力，适合作为普通高等院校、高职高专院校计算机及其相关专业的教材，也可作为从事相关工作的人员学习数据结构知识的自学教材或参考书。

本书第 1、5 章和附录由蒋秀英编写，第 2、3、4 章由燕孝飞编写，第 6、7、8 章由栾晓春编写，全书由蒋秀英策划和统稿。在本书编写过程中，课题组成员李目海、李中军、吕加国等全程参与，并给予了很好的建议。另外，在本书编写过程中，还得到了枣庄学院有关领导、信息科学与工程学院吴明君院长及同事的大力支持与帮助，在此一并表示感谢。

本书为 2009 年山东省高等学校教学改革立项项目（2009436）“基于创新能力培养的《数据结构》课程建设研究与实践”的成果之一。作者均长期从事数据结构课程教学和学生竞赛辅导工作。由于作者水平有限，书中难免有一些不足之处，在使用的过程中，有好的建议请与作者联系：sjjg@uzz.edu.cn。恳请广大读者批评、指正。

为了配合教学和参考，本书提供了配套的电子教案，读者可到中国石油大学出版社网站（http://cbs.upc.edu.cn）下载。

作　者

2011 年 7 月

目 录

CONTENTS

第1章 绪 论

本章教学重点及要求

◎ 理解数据、数据对象、数据元素、数据类型和数据结构等基本概念；
◎ 掌握数据的逻辑结构与存储结构的概念及其关系；
◎ 掌握数据的存储结构及其三个组成部分；
◎ 掌握数据的抽象数据类型和数据抽象；
◎ 掌握算法的定义、算法的特性，以及算法设计的要求；
◎ 掌握评价算法优劣的标准及方法。

1.1 数据结构的概念

计算机科学是一门研究数据表示和数据处理的科学。数据是计算机化的信息，它是计算机可以直接处理的最基本和最重要的对象。在科学计算或数据处理、过程控制以及对文件的存储和检索及数据库技术等计算机应用领域中，都是对数据进行加工处理的过程。因此，要设计出一个结构好、效率高的程序，必须研究数据的特性和数据间的相互关系及其对应的存储表示，并利用这些特性和关系设计出相应的算法和程序。

1.1.1 数据结构的重要性

用计算机处理的实际问题可分为两大类：数值计算和非数值计算。

计算机早期主要用于数值计算，涉及的运算对象是简单的整型、实型或布尔类型数据。对于这类问题，可以通过抽象出合适的数学模型，然后设计一个相应的算法来解决。所以，程序设计者的主要精力集中于“程序设计的技巧”上。

随着计算机应用领域的不断扩大，计算机更多地应用于处理非数值计算问题。这类问题涉及数据元素间复杂的相互关系，一般无法用数学方程来描述。据统计，当今处理非数值计算性的问题占用了90%以上的机器时间。

使用计算机来解决一个非数值问题一般需要经过下列几个步骤：首先要从该具体问题抽象出一个适当的数学模型，然后设计或选择一个解此数学模型的算法，最后编写程序并进行调试、测试，直至得到最终的解答，如图1-1所示。其中，寻求数学模型的实质就是分析问题，从中提取操作的对象，并找出这些操作对象之间含有的关系，然后用数学的语言加以描述。

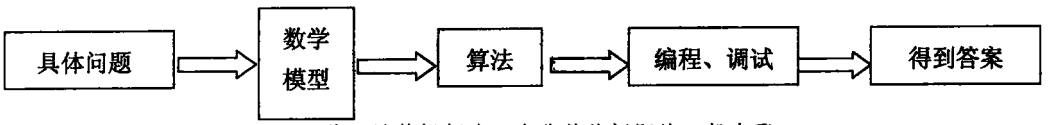


图 1-1 使用计算机解决一个非数值问题的一般步骤

因此，解决这类问题的关键不再是数学分析和计算方法，而是要设计出合适的数据结构，才能有效地解决问题。现实中对象之间的关系常有线性关系、层次关系、网状关系。下面给出这一类问题的几个实例。

【例 1-1】学生信息管理。在学生管理中，将学生记录的排列顺序按学号从小到大排列，学号较小者在前，学号较大者在后，形成一种线性关系，如表 1-1 所示。

表 1-1 学生信息表

学号	姓名	性别	出生日期	家庭地址
200712110101	陈琛军	男	1986-8-19	四川省广元市
200712110102	仇立权	男	1985-12-10	浙江省衢州市龙游县溪口镇
200712110103	崔衍丽	女	1987-8-11	安徽省宿松中学
200712110104	冬晓超	男	1986-1-2	贵州省遵义县
200712110105	甘　明	男	1986-9-13	江西省婺源县鄣山乡白石源
……	……	……	……	……

对学生信息常进行查找、插入、删除等操作。在这类数学模型中，计算机处理的对象之间通常存在一种简单的线性关系，即数据之间的 1:1 的联系。这类数学模型可称为线性的数据结构。

【例 1-2】族谱描述。图 1-2 所示为《红楼梦》中贾家的族谱，从中可以清晰地看出贾家的人物关系。在这个模型中，结点间是 1:n 的关系，是典型的层级结构，常用“树”这种存储结构来表示。“贾家”认为是树根，下面的各分支和结点是树枝和叶子，像是一棵倒立的树。可以通过遍历等操作对各结点进行访问和操作。

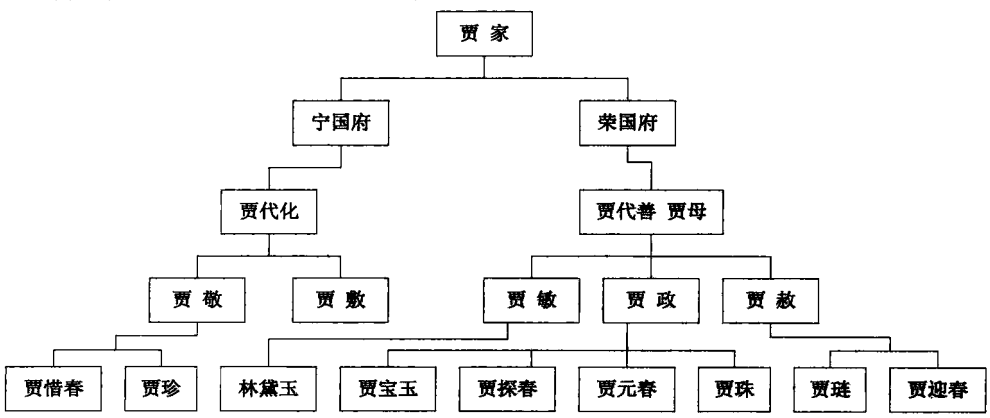


图 1-2 《红楼梦》中贾家的人物关系

【例 1-3】铺设煤气管道问题。假设要在某市的 n 个居民区之间铺设煤气管道，则在这 n 个居民区之间只要铺设 n-1 条管道即可。假设任意两个居民区之间都可以架设管道，但每条管道的建设成本不同，所要解决的问题是用一定的数据模型表示该问题，并在此基础上求投资最少（或尽可能少）的管道铺设方案。若以各小区为结点，边标注各小区间的距离为权值，则可以用图 1-3 所示的“图”结构来描述和求解这类问题。结点间为 m:n 的关系。

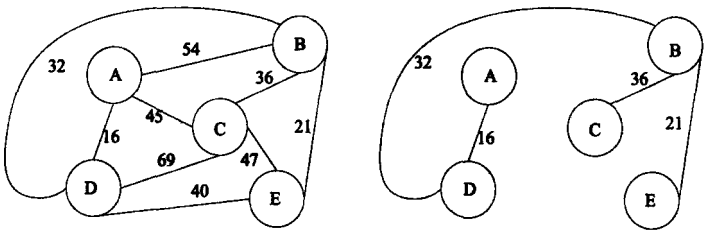


图 1-3 煤气管道铺设结构图

思考：请举出若干在现实生活中类似的线性结构、树形结构、图形结构的例子。

由以上三个例子可见，描述这类非数值计算问题的数学模型不再是数学方程，而是诸如表、树、图之类的数据结构。解决非数值计算问题的首要任务是选取一种合适的数据结构表示该问题，然后才考虑如何编写有效的算法。

因此，可以说，数据结构是一门研究非数值计算的程序设计问题中所出现的计算机操作对象以及它们之间的关系和操作的学科。

学习数据结构的目的是为了了解计算机处理对象的特性，将实际问题中所涉及的处理对象在计算机中表示出来并对它们进行处理。与此同时，通过算法训练来提高学生的思维能力，通过程序设计的技能训练来促进学生的综合应用能力和专业素质的提高。

1.1.2 基本概念和术语

在系统地学习数据结构知识之前，先对一些基本概念和术语赋予确切的含义。

1. 数据（Data）

数据是对信息的一种符号表示。在计算机科学中是指所有能输入到计算机中并被计算机程序处理的符号的总称。它可以是数值数据，也可以是非数值数据。数值数据是一些整数、实数或复数，主要用于工程计算、科学计算和商务处理等；非数值数据包括字符、文字、图形、图像、语音等。

2. 数据元素（Data Element）

数据元素是数据的基本单位，在计算机程序中通常作为一个整体进行考虑和处理。在不同的条件下，数据元素又可称为元素、结点、顶点、记录等。例如，族谱问题中的一个结点、煤气管道铺设问题中的一个顶点等，都被称为一个数据元素。

有时，一个数据元素可由若干个数据项（Data Item）组成，例如，学生信息管理系统中学生信息表的每一个数据元素就是一条学生记录，它包括学生的学号、姓名、性别、出生日期、家庭地址等数据项。数据项是数据的不可分割的最小单位。通常，在解决实际应用问题时是把每条学生记录作为一个基本单位进行访问和处理的。

3. 数据对象（Data Object）

数据对象是性质相同的数据元素的集合，是数据的一个子集。例如，整数数据对象的集合可表示为 N＝{0, ±1, ±2, …}，字母字符数据对象的集合可表示为 C={'A', 'B', …, 'Z'}。

4. 数据结构（Data Structure）

数据结构是指互相之间存在着一种或多种关系的数据元素的集合。数据元素间的相互关系包括三个方面：数据的逻辑结构、数据的存储结构、数据的运算集合。

数据的逻辑结构（Logical Structure）：从逻辑关系上描述数据，可以看作是从具体问题抽象出来的数学模型，呈现于人的外在结构，是独立于计算机的，它与数据的存储无关。根据数据元素间关系的不同特性，通常有下列四类基本的逻辑结构：

（1）集合结构。在集合结构中，数据元素间的关系是“属于同一个集合”。集合是元素关系极为松散的一种结构，可用其他结构来表示它。

（2）线性结构。该结构的数据元素之间存在着一对一的关系，第一个元素没有前驱，最后一个元素没有后继，其余的每一个元素有且只有一个前驱和后继。如表 1-1 所示的学生信息表。

（3）树形结构。该结构的数据元素之间存在着一对多的关系，最高层次的结点称为根（root）结点，只有它没有父结点（前驱），其余的每一个元素可有多个后继，但只有一个前驱。如图 1-2 所示的《红楼梦》中的贾家的族谱结构图。

（4）图形结构。该结构的数据元素之间存在着多对多的关系，元素的前驱和后继可以有多个，图形结构也称作网状结构。如图 1-3 所示的煤气管道铺设结构图。

图 1-4 为表示上述四类基本结构的示意图。

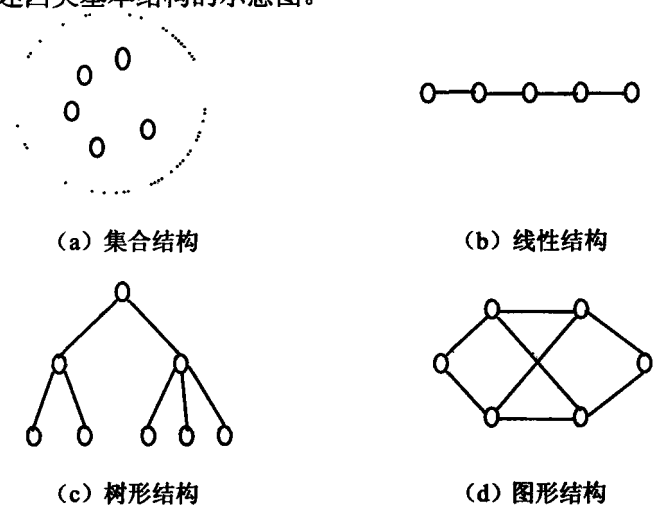


图 1-4 四类基本结构的示意图

其中，树形结构和图形结构合称为非线性结构，所以，数据结构主要分为两大类：线性结构和非线性结构（树、图）。

数据结构的逻辑结构通常可以采用一个二元组来表示，形式定义为：

$$Data\_Structure=(D, R)$$

其中，D 是数据元素的有限集，R 是 D 上关系的有限集。

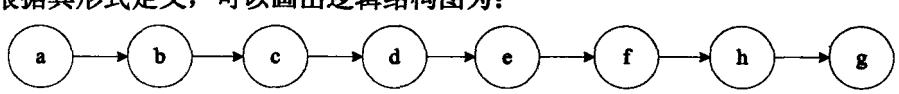
【例1-4】下列为几种用二元组表示的数据结构，画出其对应的逻辑结构图形表示，并指出属于哪一种结构。

（1）A=(D, R)，其中：

D={a,b,c,d,e,f,g,h}

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

解：根据其形式定义，可以画出逻辑结构图为：



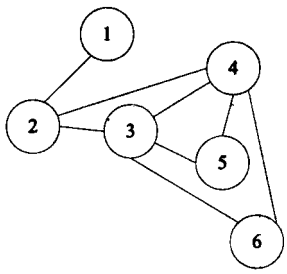
由上述可知，A属于线性结构。

（2）B=(K, R)，其中：

K={1,2,3,4,5,6}

R={(1,2),(2,3),(2,4),(3,4),(3,5),(3,6),(4,5),(4,6)}

解：根据其形式定义，可以画出逻辑结构图为：



由上述可知，B 属于图形结构。

数据的存储结构（Storage Structure）：数据结构在计算机中的表示（或称映像），又称为物理结构。它研究数据结构在计算机中的实现方法，是逻辑结构用计算机语言的实现，包括数据结构中元素的表示及元素间关系的表示。通常有四种基本的存储表示方法。

（1）顺序存储。将逻辑上相邻的结点存储在物理位置上也相邻的存储单元里，结点之间的逻辑关系由存储单元的邻接关系来表示，这种存储表示称为顺序存储结构。顺序存储结构是一种最基本的存储表示方法，通常借助于程序设计语言中的数组来实现。它主要应用于线性的数据结构。

顺序存储方法的主要优点是节省存储空间，因为分配给数据的存储单元全用于存放结点的数据，结点之间的逻辑关系没有占用额外的存储空间。采用这种方法时，可实现对结点的随机存取。主要缺点是不便于修改，对结点进行插入、删除运算时，可能要移动一系列的结点。

（2）链式存储。链式存储方法不要求逻辑上相邻的结点在物理位置上亦相邻，结点间的关系由附加的指针来表示，指针指向结点的邻接结点，这样将所有结点串联在一起，称为链式存储结构。链式存储方法不仅存储结点的值，而且还存储结点之间的关系。链式存储结构通常借助于程序设计语言中的指针类型来实现。

链式存储方法的主要优点是便于修改，在进行插入、删除运算时，仅需修改相应结点的指针域，而不必移动结点。与顺序存储方法相比，其存储空间的利用率较低，因为分配给数据的存储单元有一部分被用来存储结点之间的逻辑关系。另外，由于逻辑上相邻的结点在存储空间中不一定相邻，所以不能对结点进行随机存取。

（3）索引存储。索引存储是在存储结点信息的同时，再建立一个附加的索引表，然后利用索引表中索引项的值来确定结点的实际的存储单元地址。索引表中的每一项称为索引项，索引项的一般形式为“(关键字, 地址)”。其中，关键字能唯一标识一个结点。

这种带有索引表的存储结构可以大大提高数据查找的速度。其缺点是增加了索引表，降低了存储空间的利用率。

（4）散列（或哈希）存储。基本思想是根据结点的关键字直接计算出结点的存储地址。把结点的关键字作为自变量，通过散列函数计算规则，确定该结点的存储单元地址。

哈希存储方法的优点是查找速度快，只要给出待查结点的关键字，就可立即计算出该结点的存储地址。但与前三种存储方法不同的是，哈希存储方法只存储结点的数据，不存储结点之间的逻辑关系。哈希存储方法一般只适合要求对数据能够进行快速查找和插入的场合。

上面这四种方法既可以单独使用，也可以组合起来对数据结构进行存储。同一种逻辑结构

采用不同的存储方法，可以得到不同的存储结构。选取哪种存储结构来表示相应的逻辑结构，视具体的情况而定，主要考虑的是数据的运算是否方便及相应算法的时空复杂度。常用的是顺序存储和链式存储方法。

数据的运算是定义在数据的逻辑结构上的操作，每种逻辑结构都有一个运算的集合。最常用的运算有查找、插入、删除、更新、排序等。如在学生信息表中插入一个新生、删除退学的学生等。

数据运算是数据结构的一个重要方面，是通过算法描述来实现的，所以，讨论算法是数据结构这门课程的主要内容。任何一个算法的设计取决于选定的数据（逻辑）结构，而算法的实现依赖于采用的存储结构。

1.1.3 数据结构课程的形成和发展

20 世纪 60 年代初期，“数据结构”有关的内容散见于操作系统、编译原理和表处理语言等课程。1968 年，美国的唐·欧·克努特教授开创了“数据结构”的最初体系，他所著的《计算机程序设计技巧》中的“第一卷（基本算法）”是第一本较系统地阐述数据的逻辑结构和存储结构及其操作的著作。唐·欧·克努特教授于 1974 年获得“图灵奖”，他是 40 届中唯一一位因一部影响巨大的书而获奖的人。“数据结构”被列入美国一些大学计算机科学系的教学计划。

随后，数据结构这门学科随着计算机数据的复杂化而产生并发展起来。数据结构的概念不断扩充，包括了网络、集合代数论、关系等“离散数学结构”的内容。近年来，随着数据库系统的不断发展，在数据结构课程中又增加了文件管理（特别是大型文件的组织等）的内容。20 世纪 70 年代后期，我国高校陆续开设该课程。

数据结构的研究不仅涉及计算机硬件（特别是编码理论、存储装置和存取方法等）的研究范围，而且和计算机软件的研究有着更密切的关系，无论是编译程序还是操作系统，都涉及数据元素在存储器中的分配问题。在研究信息检索时也必须考虑如何组织数据，以便更为方便地查找和存取数据元素。

“Algorithms + Data Structures = Programs（算法+数据结构=程序）”是 1976 年瑞士的 N. Wirth 教授写的书名，由此也可看出本课程在程序设计中的重要性。

因此，可以认为，数据结构课程是介于数学、计算机硬件和计算机软件三者之间的一门核心课程，是高级程序设计语言、编译原理、操作系统、数据库、人工智能等课程的基础；亦是计算机专业本科生必修的学位课程，是计算机专业研究生入学考试的必考科目，是软件人员水平考试和计算机等级考试的内容，是 ACM 程序设计、数学建模竞赛的基础。数据结构课程所处的地位如图 1-5 所示。

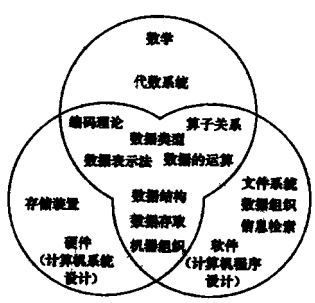


图 1-5 数据结构课程所处的地位

值得注意的是，数据结构的发展并未终结。一方面，面向各专门领域中特殊问题的数据结

构得到研究和发展，如多维图形数据结构等；另一方面，从抽象数据类型的观点来讨论数据结构，已成为一种新的趋势，越来越被人们所重视。

1.2 抽象数据类型

1.2.1 数据类型

数据类型（Data Type）：用以刻画（程序）操作对象的特性，是程序设计语言中变量所具有的数据种类。数据类型规定了在程序执行期间变量或表达式所有可能取值的范围，以及在这些值上允许进行的操作。因此，数据类型是一个值的集合和定义在这个值的集合上的一组操作的总称。例如，C 语言中的数据类型有基本类型（整型、实型、字符型和枚举类型）、指针类型、空类型、构造类型等。其中，构造类型包括数组、结构体类型、联合体类型。

在某种意义上，数据结构可以看成是“一组具有相同结构的值”，而数据类型则可被看成是由一种数据结构和定义在其上的一组操作所组成的。

1.2.2 抽象数据类型

抽象数据类型（Abstract Data Type，简称 ADT）：指的是用户进行软件系统设计时从问题的数学模型中抽象出来的逻辑数据结构和逻辑数据结构上的运算，而不考虑计算机的具体存储结构和运算的具体实现算法。可理解为对数据类型的进一步抽象，每一个操作由它的输入、输出定义。

在高级程序设计语言中，基本数据类型隐含着数据结构和定义在该结构上的操作的统一。例如，C 语言中的整型就是整数的数学含义与算术运算（+、-、*、/等）的统一体。只是由于这些基本数据类型中的数据结构的具体表示、基本操作和具体实现都很规范，所以，可以通过系统内置而隐藏起来。

对于集合、表、树、图和它们的操作一起可以看作是抽象数据类型，就像字符型、整型一样。字符型、整型有与它们相关的操作，而抽象类型也有与它们自己相关的操作。由于实际问题和问题的求解算法千变万化，所以，抽象数据类型的设计和实现不可能像基本数据类型那样规范，它要求设计人员根据实际情况具体问题具体分析，总的目标是使抽象数据类型对外的整体效率尽可能地提高。

抽象数据类型的特征是使用与实现相分离，实行封装和信息隐蔽。也就是说，在设计时，ADT 的定义取决于它的一组逻辑特性，而与其在计算机内部如何表示和实现无关。即不论其内部结构如何变化，只要它的数学特性不变，都不影响其外部的使用。抽象数据类型可以使我们更容易描述现实世界。例如，用线性表描述学生成绩表，用树描述组织结构关系等。

数据结构是 ADT 的物理实现，一般可以由数据对象、关系及操作三个要素来定义。抽象数据类型可用三元组（D, R, P）表示，其中，D 是数据对象，R 是 D 上的关系集，P 是对 D 的基本操作集。抽象数据类型定义的格式为：

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>
基本操作：<基本操作的定义>
} ADT 抽象数据类型名

其中，数据对象和数据关系的定义用伪码描述，基本操作的定义格式为：

基本操作名(参数表)
初始条件：<初始条件描述>
操作结果：<操作结果描述>

【例 1-5】抽象数据类型复数的定义为：

```
ADT Complex
{
    数据对象：
        D={i1,i2|i1，i2 均为实数}
    数据关系：
        R1={<i1,i2>| i1 是复数的实数部分，i2 是复数的虚数部分}
    基本操作：
        AssignComplex(&Z,v1,v2)：构造复数 Z，其实部和虚部分别赋给参数 v1 和 v2。
        DestroyComplex(&Z)：复数 Z 被销毁。
        GetReal(Z,&real)：用 real 返回复数 Z 的实部值。
        GetImag(Z,&Imag)：用 Imag 返回复数 Z 的虚部值。
        Add(z1,z2,&sum)：用 sum 返回两个复数 z1、z2 的和值。
}ADT Complex
```

1.3 算法和算法分析

算法与数据结构的关系紧密，在算法设计时先要确定相应的数据结构，而在讨论某一种数据结构时也必然会涉及相应的算法。下面就从算法特性、算法设计的要求、算法性能分析与度量等三个方面对算法进行介绍。

1.3.1 算法的特性

算法（Algorithm）是对特定问题求解步骤的一种描述，是指令的有限序列，其中，每一条指令表示一个或多个操作。

一个算法应该具有下列五个特性：

（1）有穷性：一个算法必须在有穷步之后结束，且每一步都在有穷时间内完成。

（2）确定性：算法的每一条指令必须有确切的含义，不存在二义性，且算法只有一个入口和一个出口。

（3）可行性：算法描述的操作都是可以通过已经实现的基本运算执行有限次来实现的。

（4）输入：一个算法具有零个或多个输入，这些输入取自于某个特定的对象集合。

（5）输出：一个算法具有一个或多个输出，输出同输入之间存在某种特定的关系。

注意：经常以填空题、选择题的形式考查本内容。

算法的含义与程序十分相似，但二者是有区别的。一个程序不一定满足有穷性（死循环），另外，程序中的指令必须是机器可执行的，而算法中的指令则无此限制。一个算法若用计算机语言来书写，则它就可以是一个程序。

算法与数据结构紧密相关，数据结构是算法的基础，直接影响算法的效率。反之，一种数据结构的优劣由各种算法的执行效率来体现。

1.3.2 算法设计的要求

要设计一个好的算法，通常要考虑以下要求：

（1）正确性（Correctness）。算法的执行结果应当满足预先规定的功能和性能的要求。要求算法的编写者对问题的要求有正确的理解，并能正确、无歧义地描述和利用某种编程语言正确地实现算法。这是最重要的标准。

（2）可读性（Readability）。一个算法应当思路清晰、层次分明、简单明了、易读易懂。在算法中必须加入注释，简要说明算法的功能、输入与输出参数的使用规则、重要数据的作用和算法中各程序段完成的功能等。这是理解、测试和修改算法的需要。

（3）健壮性（Robustness）。当输入不合法数据时，算法应能作适当处理，不致引起严重后果。要求算法做容错性或例外处理，能够对不合理的数据进行检查。

（4）效率与低存储量需求。效率指的是算法的执行时间。对于同一个问题，如果有多个算法可以解决，执行时间短的算法效率高。存储量需求是指算法执行过程中所需要的最大存储空间。一般来说，这两者与问题的规模有关。

1.3.3 算法的性能分析与度量

同一问题可用不同的算法来解决，而一个算法的质量优劣将影响到算法乃至程序的效率。算法分析的目的在于选择合适的算法和改进算法。一般从时间复杂度与空间复杂度两方面来评价算法的优劣。

1. 时间复杂度

当我们将一个算法转换成程序并在计算机上执行时，其运行所需要的时间取决于下列因素：

（1）依据的算法选用何种策略。

（2）书写程序的语言。实现语言的级别越高，其执行效率就越低。

（3）编译程序所生成目标代码的质量。对于代码优化较好的编译程序，其所生成的程序质量较高。

（4）问题的规模。例如，求 100 以内的素数与求 1000 以内的素数，其执行时间必然是不同的。

（5）机器执行指令的速度。

显然，在各种因素都不能确定的情况下，很难比较出算法的执行时间。为此，可以认为一个特定算法的运行工作量的大小就只依赖于问题的规模（通常用正整数 n 表示），或者说它是问题规模的函数。

一个算法是由控制结构（顺序、分支和循环三种）和原操作（指固有数据类型的操作）构