



普通高等学校“十一五”规划教材

数据结构



主 编 陈广山 郭 群 冯素梅

本书配有
电子教案



国防工业出版社
National Defense Industry Press

普通高等学校“十一五”规划教材

数据结构

主编 陈广山 郭群 冯素梅

国防工业出版社

· 北京 ·

内 容 简 介

“数据结构”是计算机与信息技术类专业的一门专业基础课,是从事计算机与信息技术工作人员的必修课程之一。本书涵盖了教育部对“数据结构”教学规定的全部内容和全国硕士研究生入学考试“数据结构”大纲中规定的知识点,并融入了编者多年教学经验的总结和对信息技术行业背景知识的进一步思考。全书共分10章,内容涉及数据结构的基本概念、线性表、栈和队列、串、数组与广义表、树、图、查找、排序以及文件等。

本书可作为高等院校计算机与信息技术类专业的本、专科的教材和参考书,也可供从事计算机与信息技术的工程人员参考(如需电子教学资源,请发邮件至 lyzhang@ndip.cn 索取)。

图书在版编目(CIP)数据

数据结构/陈广山主编. —北京:国防工业出版社,
2009.1
普通高等学校“十一五”规划教材
ISBN 978-7-118-06029-4

I. 数... II. 陈... III. 数据结构—高等学校—教材 IV. TP311.12

中国版本图书馆 CIP 数据核字(2008)第 191884 号

※

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100048)

腾飞印务有限公司印刷
新华书店经售

※

开本 787×1092 1/16 印张 18½ 字数 424 千字

2009 年 1 月第 1 版第 1 次印刷 印数 1—4000 册 定价 30.00 元

(本书如有印装错误,我社负责调换)

国防书店: (010)68428422

发行邮购: (010)68414474

发行传真: (010)68411535

发行业务: (010)68472764

前 言

随着计算机应用的普遍,计算机在非数值计算领域的应用越来越广泛,而数据结构知识是解决这类问题的基础,所以,“数据结构”在计算机与信息技术类专业的教学中具有突出的地位和作用。

“数据结构”是计算机与信息技术类专业的一门重要的专业基础课程,也是该专业的一门核心关键性课程。本书系统地介绍了常用的数据结构以及相应的算法,包括数据结构的基本概念、线性表、栈和队列、串、数组与广义表、树、图、查找和排序等。在计算机与信息科学中,数据结构不仅是一般程序设计的基础,还是设计和实现编译程序、操作系统、数据系统及其他系统程序的重要基础,本课程将为后续课程的学习、程序设计效率的提高奠定良好的基础。

本书是根据教育部有关“数据结构”教学的建议,以及全国计算机专业硕士研究生入学考试“数据结构”大纲编写的。在编写过程中,以通俗易懂、简明实用为宗旨,以方便教师的教和学生的学为基本理念,以基本知识为基础,重点体现理论够用、实践实用的原则,注意通过典型的案例促进对各种数据结构的理解。本书在编写过程中也注意了语言的形象性,避免晦涩现象的出现;力求抽象与形象相结合,知识点与计算机领域的具体应用相联系;算法与程序、习题相对应。

全书共分 10 章:第 1 章是绪论,主要介绍了数据结构的相关概念和术语,以及算法的特点和评价标准等内容;第 2 章介绍了线性表的有关概念和基本操作;第 3 章对栈和队列的知识进行了重点介绍;第 4 章为数组和广义表,对数组和广义表的基本表示和实现进行了研究;第 5 章的主要内容是串的基本操作;第 6 章介绍了典型的非线性结构二叉树,包括树、森林的基本概念、表示与实现,以及哈夫曼树的应用等内容;第 7 章介绍了图的基本理论,并对最短路径、关键路径、拓扑排序等应用做了重点介绍;第 8 章主要介绍了几种传统的查找方法,并对散列查找等现代查找方法进行了详细介绍。第 9 章主要介绍了常用的排序方法;第 10 章简单地介绍了文件的知识和基本操作。

为了加强对知识点的理解,每一章的后面都选配了适量的习题,供读者参考。为了方便教师开展教学工作,本书提供电子教案、课件和习题等教学资源。此外,本书还设计了相应的实验,有需要者请发邮件至 lyzhang@ndip.cn 索取。

本书由陈广山、郭群、冯素梅编写,其中第 6 章、第 8 章、第 9 章由郭群编写,第 3

章~第5章、第10章由冯素梅编写,第1章、第2章、第7章由陈广山编写。

本书的编者一直在教学一线从事“数据结构”的教学工作,在编写的过程中融入了编者的教学经验和对信息技术行业背景知识的进一步理解。本书可作为高等院校计算机与信息技术类专业本、专科教学的教材和参考书,也可供从事计算机与信息技术的工程人员参考。在教学的过程中可根据实际情况对书中的内容进行取舍,参考学时为54学时~72学时。

由于编者水平所限,本书难免存在疏漏或不妥之处,恳请读者和同行赐教。

编者

2008年10月

目 录

第 1 章 绪论	1
1.1 数据结构的概念	1
1.1.1 计算机解决问题的步骤	1
1.1.2 基本概念和术语	4
1.1.3 研究数据结构的意义	6
1.2 抽象数据类型的表示	7
1.2.1 数据类型	7
1.2.2 抽象数据类型	7
1.2.3 抽象数据类型的表示	8
1.3 算法和算法分析	10
1.3.1 算法特性	10
1.3.2 算法描述	11
1.3.3 算法性能分析与度量	11
1.4 本章小结	13
习题	14
第 2 章 线性表	17
2.1 线性表的定义及其基本操作	17
2.1.1 线性表的定义	17
2.1.2 线性表的基本操作	17
2.2 线性表的顺序表示与实现	19
2.2.1 顺序表	19
2.2.2 顺序表基本运算的实现	20
2.3 线性表的链式表示与实现	24
2.3.1 单链表	24
2.3.2 单链表的基本运算	25
2.4 双向链表	28
2.4.1 双向链表的存储结构	28
2.4.2 双向链表的基本运算	29
2.5 循环链表	30
2.6 静态链表	31

2.7 线性表的应用·····	33
2.8 顺序表和链表的比较·····	36
2.9 本章小结·····	36
习题·····	37
第3章 栈和队列 ·····	41
3.1 栈·····	41
3.1.1 栈的抽象数据类型的定义·····	41
3.1.2 栈的表示与实现·····	42
3.2 队列·····	46
3.2.1 队列的抽象数据类型的定义·····	46
3.2.2 队列的表示与实现·····	47
3.3 栈和队列的应用·····	53
3.3.1 栈的应用·····	53
3.3.2 队列的应用·····	62
3.4 本章小结·····	65
习题·····	65
第4章 数组和广义表 ·····	67
4.1 数组的概念·····	67
4.2 数组的存储结构·····	68
4.2.1 数组的内存映像·····	68
4.2.2 数组的顺序存储表示与实现·····	69
4.3 矩阵的压缩存储·····	71
4.3.1 特殊矩阵·····	72
4.3.2 稀疏矩阵·····	74
4.4 广义表的概念·····	84
4.5 广义表的存储结构·····	86
4.6 数组和广义表的应用·····	90
4.6.1 数组的应用·····	90
4.6.2 广义表的应用·····	93
4.7 本章小结·····	95
习题·····	95
第5章 串 ·····	98
5.1 串的基本概念·····	98
5.2 串的实现·····	100
5.2.1 串的顺序存储表示与实现·····	100
5.2.2 串的链式存储表示与实现·····	105

5.3 模式匹配	107
5.3.1 简单的模式匹配算法	107
5.3.2 KMP 算法	109
5.4 串的应用	114
5.4.1 文本编辑	114
5.4.2 文本加密	116
5.4.3 文本替换	118
5.5 本章小结	118
习题	119
第 6 章 树和二叉树	121
6.1 树	121
6.1.1 树的基本概念	121
6.1.2 树的表示	124
6.2 二叉树	125
6.2.1 二叉树的概念	125
6.2.2 二叉树的性质	127
6.2.3 二叉树的存储结构	128
6.3 二叉树的遍历	131
6.3.1 二叉树的遍历方法及递归实现	131
6.3.2 二叉树遍历的非递归实现	133
6.3.3 二叉树的应用	136
6.3.4 二叉树遍历的应用	137
6.4 线索二叉树	140
6.4.1 线索二叉树的定义及结构	140
6.4.2 线索二叉树的基本操作实现	142
6.5 树和森林	145
6.5.1 树的存储结构	145
6.5.2 树、森林与二叉树的关系	147
6.5.3 树和森林的遍历	149
6.6 哈夫曼树及其应用	150
6.6.1 哈夫曼树	150
6.6.2 哈夫曼树编码	153
6.6.3 哈夫曼树的应用	155
6.7 树的应用	157
6.7.1 集合的表示	157
6.7.2 关系等价求等价类问题	159
6.8 本章小结	160
习题	161

第 7 章 图	164
7.1 图的基本概念	164
7.1.1 图的定义	164
7.1.2 图的基本术语	165
7.2 图的存储结构	168
7.2.1 邻接矩阵	168
7.2.2 邻接表	170
7.2.3 十字链表	172
7.2.4 邻接多重表	174
7.3 图的遍历	176
7.3.1 深度优先遍历	176
7.3.2 广度优先遍历	178
7.4 最小生成树	179
7.4.1 最小生成树的概念	179
7.4.2 构造最小生成树的 Prim 算法	179
7.4.3 构造最小生成树的 Kruskal 算法	182
7.5 最短路径	184
7.5.1 从一个源点到其他各顶点的最短路径	184
7.5.2 每一对顶点之间的最短路径	187
7.6 有向无环图及其应用	189
7.6.1 有向无环图的概念	189
7.6.2 拓扑排序	189
7.6.3 关键路径	194
7.7 图的应用	198
7.7.1 图在路由器寻径中的应用	199
7.7.2 图在物流信息系统中的应用	199
7.8 本章小结	200
习题	200
第 8 章 查找	204
8.1 查找的基本概念	204
8.2 静态查找表	206
8.2.1 顺序表的查找	206
8.2.2 有序表的查找	208
8.2.3 索引表的查找	211
8.3 动态查找表	211
8.3.1 二叉排序树和平衡二叉树	212
8.3.2 B-树和 B+ 树	224

8.4 哈希表	230
8.4.1 什么是哈希表	230
8.4.2 哈希函数的构造方法	232
8.4.3 冲突的解决方法	234
8.4.4 哈希表的查找	236
8.5 查找的应用	238
8.5.1 在 ARM Linux 中的应用	238
8.5.2 病毒特征代码串的搜索	239
8.6 本章小结	241
习题	241
第 9 章 排序	245
9.1 排序的基本概念	245
9.2 内部排序	246
9.2.1 插入排序	246
9.2.2 快速排序	252
9.2.3 选择排序	256
9.2.4 归并排序	260
9.2.5 基数排序	262
9.2.6 各种内部排序方法的比较和选择	265
9.3 外部排序	266
9.3.1 外部排序的方法	267
9.3.2 多路平衡归并的实现	268
9.4 本章小结	270
习题	271
第 10 章 文件	273
10.1 文件的基本概念	273
10.2 文件组织	274
10.3 本章小结	280
习题	280
附录 A 习题参考答案(部分)	282
附录 B 2009 年全国计算机专业硕士研究生入学考试大纲(数据结构)	284
参考文献	286

第1章 绪 论

21 世纪是一个信息文明的新世纪，计算机已经渗透到人类生活的每一个角落，计算机信息处理已经成为人类活动不可缺少的组成部分。计算机科学是一门研究信息表示和处理的科学，数据是计算机可以直接处理的最基本和最重要的对象。随着计算机科学技术的迅速发展，计算机的应用已经从传统的数值计算领域发展到各种非数值计算领域，处理的对象扩展到字符、表格、声音、图像和动画等多媒体信息。信息处理也不再是单纯的计算，而是诸如信息存储与检索等非数值计算问题。

因此，如何在计算机中表示数据、如何有效地存取数据以及如何有效地实现数据对象之间的关系是人类必须解决的新问题。传统的解决数值计算问题的理论、方法和技术已经无法满足非数值计算问题的需要，必须建立新的理论、方法和技术体系，“数据结构”就是在解决前述问题的过程中诞生的一门新学科。

1.1 数据结构的概念

“数据结构”是计算机与信息技术类专业的专业基础课，是一门十分重要的核心课程。所有的计算机系统软件和应用软件都要用到各种类型的数据结构。因此，要想更好地运用计算机来解决问题，仅有计算机程序设计语言是不够的。要想有效地使用计算机、充分发挥计算机的性能，还必须学习和掌握好数据结构的有关知识和基本技能。此外，“数据结构”对学习计算机专业的其他课程，如操作系统、编译原理、软件工程、人工智能等都具有十分重要的意义。

1.1.1 计算机解决问题的步骤

一般来说，用计算机解决问题时大致经过如下几步：首先从具体问题中抽象出一个适当的数学模型；然后选择或设计一个解此数学模型的数据结构和算法；最后编出程序、调试、测试，直至得到最终结果。

1. 寻求数学模型

寻求数学模型的实质就是分析、总结、提取操作的对象，并找出这些操作对象之间的关系，然后用数学语言加以描述。

(1) 分析问题：分析问题是寻求从数据输入到输出的过程。

(2) 进行抽象：抽象就是抽取问题的实质，而忽略其细节，形成数学模型。抽象有过程抽象和数据抽象两种形式：过程抽象是面向过程的程序设计经常使用的方法，它将整个功能划分为若干个部分，强调功能完成的过程和步骤；数据抽象是面向对象程序设计所使用的方法，是指将要处理的数据和这些数据上的操作进行捆绑，从而形成具有不同

的功能和性质的抽象数据类型。

2. 确定数据结构

在完成了对问题的抽象工作以后，就要选取或设计合适的数据结构来表示这个问题的数据及数据之间的关系。为此，需要研究各种典型的数据结构的特点及其适用的范围。

3. 算法设计

一旦数据结构得以确定，接下来的工作就是进行算法设计了。解决一个问题通常会有多种算法可供选择，关键在于针对具体问题选择合适的算法。如果没有完全适用的算法，则需要根据具体问题的特点和所确定的数据结构设计出相应的算法。同时，需要对算法进行分析、评估，并建立一套算法性能分析和评估的体系和方法。

4. 程序设计

对于待解决的具体问题来说，经过算法分析并确定了算法之后，就可以选用一种计算机程序设计语言编写相应的程序，经过调试并运行这个程序，从而给出问题的解。

如果所得的结论符合问题的要求，则这个算法是有效的；反之，则需要重新检查，修正上述步骤，直到问题被合理解决为止。

对于数值问题，其所涉及的运算对象主要是整型、实型或布尔类型等简单的数据类型，所以程序设计者的主要精力是放在程序设计的技巧上，无需重视数据结构。但对于非数值计算问题，由于数据元素的相互关系一般无法用数学方程式加以描述，因此，解决这类问题的关键也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，才能有效地解决之。请研究下面所述的3个问题。

例 1-1 大学教务管理信息系统(MIS)中学生信息的检索问题。如果需要查找某个学生的有关信息，或者想查询某个专业或年级学生的相关信息，只要建立了合适的数据结构，并按照其所对应的算法编写相关程序，就可以实现学生相关信息的计算机自动检索了。

通过“数据库原理”课程的学习，对于这类问题可以由一个按学号顺序排列的“学生信息表”和分别按姓名、专业、年级顺序排列的“索引表”来实现，如图 1-1 所示。由这 4 张表构成的数据库文件便是学生信息检索的数学模型，计算机的主要操作就是通过某个特定“关键字”的比较，从而实现对学生的信息文件的查询。

学号	姓 名	性 别	专 业	年 级
050001	赵铭志	男	信息管理与信息系统	05 级
050002	李小芳	女	信息与计算科学	05 级
060301	刘丽丽	女	软件技术	06 级
060302	张学友	男	信息与计算科学	06 级
060303	李宝国	男	计算机科学与技术	06 级
070801	邓文羨	女	软件技术	07 级
070802	陈利国	男	软件技术	07 级
070803	方明靖	男	信息管理与信息系统	07 级
070804	王 丽	女	计算机科学与技术	07 级

(a)

陈利国	7
邓文菱	6
方明靖	8
李宝国	5
李小芳	2
刘丽丽	3
王丽	9
张学友	4
赵铭志	1

(b)

信息管理与信息系统	1, 8
信息与计算科学	2, 4
软件技术	3, 6, 7
计算机科学与技术	5, 9

(c)

05 级	1, 2
06 级	3, 4, 5
07 级	6, 7, 8, 9

(d)

图 1-1 学生信息管理系统中的数据结构

(a) 学生信息表; (b) 姓名索引表; (c) 专业索引表; (d) 年级索引表。

诸如此类的问题还有图书馆的书目检索系统、考试查分系统、企业进销存管理系统等。在这类数学模型中，计算机处理的对象之间通常存在着的是一种简单的线性关系，这类数学模型可称为线性的数据结构。

例 1-2 Internet 域名(或计算机文件系统)管理问题。在“计算机网络”课程中，Internet 的域是一个树形结构，仅用数值计算问题的解决策略是不可行的，如图 1-2 所示。在 Internet 资源的搜索问题中，处理过程不是根据某种确定的计算法则，而是一个按照某种规则遍历树的过程。在这类问题中所出现的树也是一种数据结构，它可以应用在许多非数值计算的问题中，如计算机博弈问题、计算机决策等问题中所涉及的数据结构都是树形结构。

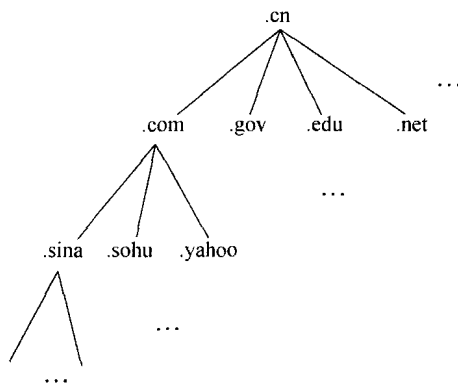


图 1-2 Internet 域名组织中的数据结构

例 1-3 火车交通咨询系统问题。火车交通咨询系统对乘客来说是一个非常实用的计算机应用系统，它不但可以提供详细的换乘车情况，而且还可以为乘客从多条可行的路径中选择一个“最佳”的方案。设计一个计算机交通咨询系统首要的任务是解决数据的表示问题。如图 1-3 所示，通常用顶点表示城市，顶点间的连线表示两个城市间存在的一条路径。在此数据结构的基础上，按照某种策略通过检测顶点和边的关系就可以解决乘客关心的问题。

这种城市之间的关系可用一个称为图的数据结构来表示，图的应用也是典型的非数值计算的问题，如路由器中的路由算法、工程中的关键路径计算问题等都是图形结构。

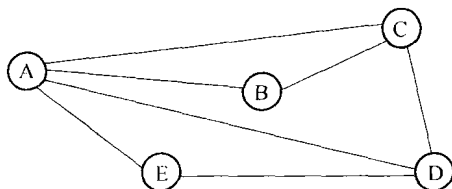


图 1-3 火车交通咨询系统中的数据结构

由以上 3 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。因此，可以说数据结构主要是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。

1.1.2 基本概念和术语

1. 数据

数据(Data)是信息的载体，是计算机程序加工的原料，能够被计算机识别、存储和加工处理。计算机科学中，所谓数据就是计算机加工处理的对象，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

数据元素(Data Element)是数据的基本单位。在不同的条件下，数据元素又可称为元素、结点、顶点、记录等。例如，数据库表中的一个记录、树型结构中的一个结点、图形结构中的一个顶点等，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，学生信息表的每一个数据元素就是一个学生记录。它包括学生的学号、姓名、性别、籍贯、出生年月、成绩等数据项。数据项是在数据处理时不能再分割的最小单位。通常，在解决实际问题时是把每个记录当作一个基本单位进行访问和处理的。

3. 数据对象

数据对象(Data Object)或数据元素类(Data Element Class)是性质相同的数据元素的集合。在某个具体问题中，数据元素都具有相同的性质，属于同一数据对象(数据元素类)，数据元素是数据元素类的一个实例。例如，在交通咨询系统的交通网中，所有的顶点是一个数据元素类，顶点 A 和顶点 B 各自代表一个城市，是该数据元素类中的两个实例，其数据元素的值分别为 A 和 B。

4. 数据结构

数据结构(Data Structure)是指互相之间存在着一种或多种关系的数据元素的集合。在任何问题中，数据元素之间都不会是孤立的，在它们之间都存在着这样或那样的关系，这种数据元素之间的关系称为结构。根据数据元素间关系的不同特性，通常有下列 4 类基本的结构：

(1) 集合结构。在集合结构中，数据元素之间只是“属于同一个集合”而已，而无其他关系。集合是元素关系极为松散的一种结构。

(2) 线性结构。该结构的数据元素之间存在着一对一的关系。

(3) 树形结构。该结构的数据元素之间存在着一对多的关系。

(4) 图形结构。该结构的数据元素之间存在着多对多的关系，带权的图形结构也称为网状结构。图 1-4 为上述 4 类基本结构的示意图。

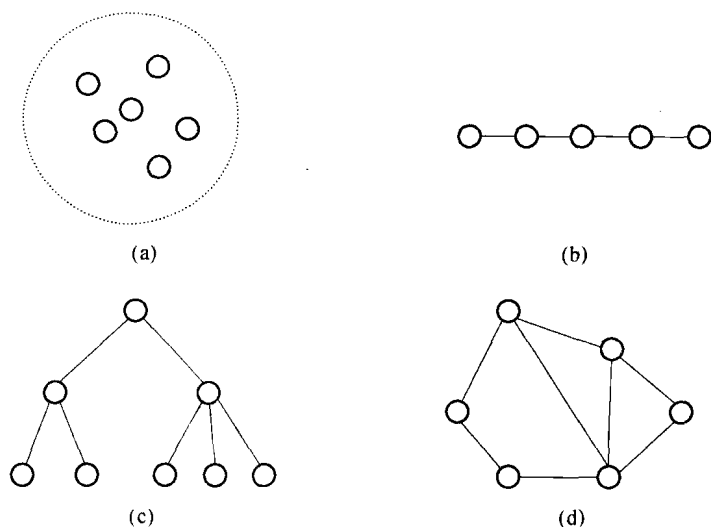


图 1-4 4 类基本结构的示意图

(a) 集合结构；(b) 线性结构；(c) 树形结构；(d) 图形结构。

其中集合是数据元素之间关系极为松散的一种结构，也可以用其他的结构来表示，因此本书就不赘述，而把重点放在其他 3 种数据结构的介绍上。

从以上数据结构的概念中可以知道，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。在形式上，数据结构通常可以采用一个二元组来表示。

数据结构的形式定义：数据结构是一个二元组，即

$$\text{Data_Structure} = (D, R) \quad (1-1)$$

式中：D 为数据元素的有限集；R 为 D 上关系的有限集。

5. 逻辑结构和物理结构

数据结构包括数据的逻辑结构和数据的物理结构。数据的逻辑结构可以看作是从具体问题抽象出来的数学模型，它与数据的存储无关。研究数据结构的目的是为了在计算机中实现对它的操作，为此还需要研究如何在计算机中表示一个数据结构。数据结构在计算机中的标识(又称映像)称为数据的物理结构，或称存储结构。它所研究的是数据结构在计算机中的实现方法，包括数据结构中元素的表示及元素间关系的表示。

6. 顺序存储和链式存储

数据的存储结构通常采用顺序存储或链式存储的方法。

顺序存储方法是把逻辑上相邻的元素存储在物理位置相邻的存储单元中，由此得到的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，通常借助程序设计语言中的数组来实现。

链式存储方法对逻辑上相邻的元素不要求其物理位置相邻，元素间的逻辑关系通过附设的指针域来表示，由此得到的存储表示称为链式存储结构，链式存储结构通常借助

程序设计语言中的指针类型来实现。

除了通常采用的顺序存储方法和链式存储方法外，有时为了查找的方便还采用索引存储方法和散列存储方法。

1.1.3 研究数据结构的意义

“数据结构”作为一门独立的课程在国外是从 1968 年才开始的，但在此之前其有关内容已散见于“编译原理”及“操作系统”等课程之中。20 世纪 60 年代中期，美国的一些大学开始设立有关课程，但当时的课程名称并不叫“数据结构”。1968 年美国唐·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》第 1 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。从 20 世纪 60 年代末到 70 年代初，出现了大型程序，软件也相对独立，结构程序设计成为程序设计方法学的主要内容，人们越来越重视数据结构。从 70 年代中期到 80 年代，各种版本的数据结构著作相继出现。目前，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，从抽象数据类型和面向对象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。

“数据结构”课程集中讨论了软件开发过程中的设计阶段、代码设计和分析阶段的若干基本问题。此外，为了构造出好的数据结构及其实现，还需考虑数据结构及其实现的评价与选择。因此，数据结构的内容包括 3 个层次的 5 个要素，如图 1-5 所示。

层次	数据加工过程	
	数据表示	数据处理
抽象	逻辑结构	基本运算
实现	存储结构	算法
评价	不同数据结构的比较及算法分析	

图 1-5 “数据结构”课程内容体系

数据结构的核⼼技术是分解与抽象。通过分解可以划分出数据的 3 个层次；再通过抽象，舍弃数据元素的具体内容，就得到逻辑结构。类似地，通过分解将处理要求划分成各种功能，再通过抽象舍弃实现细节，就得到运算的定义。上述两个方面的结合可以将具体问题变换为数据结构。这是一个从具体(具体问题)到抽象(数据结构)的过程。然后，通过增加对实现细节的考虑，进一步得到存储结构和实现运算，从而完成设计任务。这是一个从抽象(数据结构)到具体(具体实现)的过程。

数据结构与数学、计算机硬件和软件有着十分密切的关系。数据结构是介于数学、计算机硬件和计算机软件之间的一门计算机与信息技术类专业的核⼼课程。在分析和开发计算机系统与应用软件中都要用到数据结构知识，数据结构既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，也是开发好软件的必备知识。目前，数据结构技术正广泛地应用于信息科学、系统工程以及各种工程技术领域之中，众所周知，任何一个程序都是建立和运行在相应的数据结构基础之上的，建立在合理数据结构之上的算法(程序)才能做到正确、快速和高效。熟练地掌握数据结构知识和技能是 IT 专业人才所必需的素质。

1.2 抽象数据类型的表示

1.2.1 数据类型

数据类型(Data Type)和数据结构是一个密切相关的概念。它最早出现在高级程序设计语言中,用以描述程序中操作对象的特性。在用高级语言编写的程序中,每个变量、常量或表达式都有一个它所属的确定的数据类型。类型显式或隐含地规定了在程序执行期间变量或表达式所有可能的取值范围,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集上的一组操作的总称。

在高级程序设计语言中,数据类型可分为两类: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,如C语言中整型、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,并且它的成分可以是非结构的,也可以是结构的,例如,数组的值由若干分量组成,每个分量可以是整数,也可以是数组等。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, ADT)是指一个数学模型以及定义在该模型上的一组操作。抽象数据类型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。

抽象数据类型和数据类型实质上是一个概念。例如,各种计算机都拥有的整数类型就是一个抽象数据类型,尽管它们在不同处理器上的实现方法可以不同,但由于其定义的数学特性相同,在用户看来都是相同的。因此,“抽象”的意义在于数据类型的数学抽象特性。

但在另一方面,抽象数据类型的范畴更广,它不再局限于前述各处理器中已定义并实现的数据类型,还包括用户在设计软件系统时自己定义的数据类型。为了提高软件的重用性,在近代程序设计方法学中,要求在构成软件系统的每个相对独立的模块上,定义一组数据和施于这些数据上的一组操作,并在模块的内部给出这些数据的表示及其操作的细节,而在模块的外部使用的只是抽象的数据及抽象的操作。这也就是面向对象的程序设计方法。

抽象数据类型的定义可以由一种数据结构和定义在其上的一组操作组成,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种要素来定义。可表示为

$$ADT=(D, S, P) \quad (1-2)$$

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。

抽象的数据类型通常用以下格式定义:

ADT 抽象数据类型名

{数据对象: <数据对象的定义>