



浩强工作室  
HAO QIANG STUDIO

21 世纪计算机基础教育系列教材

谭浩强 主编

# 实用数据结构教程

赵丹亚 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

数据结构主要研究数据间的联系(数据的逻辑结构),数据在计算机中的存储方法(数据的物理结构),以及处理不同结构数据的算法。本教材共分6章。第1章着重介绍了数据结构课程的背景、研究对象和本教材中算法的描述和分析方法。第2章到第5章以线性结构、层次结构、网状结构为主线,由简到繁地介绍了顺序表、链表以及栈和队,树和二叉树,图等几种基本数据结构及有关算法。第6章较为详细地讨论了文件的索引技术以及常用文件的组织方法。本教材大多数内容都给出了PAD图描述的算法,读者很容易阅读理解和分析,一旦需要,可以方便地转换成任何一种结构化程序设计语言上机执行。

本教材的主要对象是经济信息管理专业或其他计算机应用专业的大学本科、专科学生,亦可供从事计算机应用工作的管理人员和技术人员学习参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有,侵权必究。

### 图书在版编目(CIP)数据

实用数据结构教程/赵丹亚编著. —北京:电子工业出版社,2002.7  
21世纪计算机基础教育系列教材  
ISBN 7-5053-7713-2

I. 实… II. 赵… III. 数据结构—教材 IV. TP311.12

中国版本图书馆CIP数据核字(2002)第039772号

责任编辑:冉 哲

印 刷:北京四季青印刷厂

出版发行:电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路173信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:9.25 字数:236.8千字

版 次:2002年7月第1版 2002年7月第1次印刷

印 数:3000册 定价:14.00元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。  
联系电话:(010)68279077

## 《21 世纪计算机基础教育系列教材》序

21 世纪是信息时代,是科学技术高速发展的时代。计算机技术与网络技术的结合,使人类的生产方式、生活方式和思维方式发生了深刻的变化。在新世纪中,计算机知识已成为当代人类文化的一个重要组成部分了。我们要把计算机知识和应用向一切有文化的人普及。

高等学校承担着为社会培养高层次人才的任务,大学生毕业后应当成为我国各个领域中的计算机应用人才,成为向全社会推广计算机应用的积极分子。在大学里应当把计算机教育放在十分重要的位置。

我国高校的计算机基础教育起步于 20 世纪 80 年代初。20 年来从无到有地迅速发展,从理工科专业发展到所有专业,从最初只开设一门语言课到按三个层次设置课程,学时也从三四十小时增加到一二百小时。计算机基础教育已经先后上了几个台阶,现在又需要上一个新的台阶。在新世纪初,我们要求提高大学生应用计算机的能力,以适应科学技术和经济发展的需要。

我们在这里所说的计算机基础教育,是指面对全体大学生的计算机教育。非计算机专业和计算机专业中的计算机教育的特点有很大的区别。无论学生基础、培养目标、教学要求、教学内容、教学方法和教材,都和计算机专业有很大的不同。绝不可简单地照搬计算机专业的模式,否则必事倍功半。计算机基础教育实际上是计算机应用的教育,应当以应用为目的,以应用为出发点。

计算机不仅是一种工具,也是一种文化,工具是可选的,文化却是必备的。对学生来说,它还是全面素质教育的一个重要部分,通过学习计算机知识能激发学生对先进科学技术的向往,启发学生对新知识的学习热情,培养学生的创新意识,提高学生的自学能力,锻炼学生动手实践的能力。多年来的实践证明,对计算机感兴趣的学生,绝大多数都是兴趣广泛、思想活跃、善于思考、自学能力较强、喜欢动手实践的。他们绝不是只会死背书本的书呆子。

我们必须认真分析非计算机专业的特点,根据教学上的需要与可能,制定出恰当的教学要求,使学生在有限的时间内能学到最多的有用的知识。全国高等院校计算机基础教育研究会曾提出了在计算机基础教育中应当正确处理的 10 个关系,即:(1)理论与应用的关系,(2)深度与广度的关系,(3)当前与发展的关系,(4)硬件与软件的关系,(5)追踪先进水平与教学相对稳定的关系,(6)课内与课外的关系,(7)课程设置与统一考试的关系,(8)计算机课程与其他课程的关系,(9)要求学生动手能力强与当前设备不足的矛盾,(10)计算机技术发展迅速与师资现状的矛盾。在教学实践中,许多学校都创造了丰富的经验。

在非计算机专业的教学中,首先需要解决的问题是:准确定位,合理取舍教学内容。我们必须分清楚:哪些内容是需要,哪些内容是不需要的;哪些内容是目前暂时可以不学而留待以后学的,哪些内容是目前不必学而以后也不必学的;哪些内容是主要的,哪些是次要的。绝不可眉毛胡子一把抓,不加分析、不问主次,使学生感到难以入门。

在教学方法和教材的编写上,要善于用通俗易懂的方法和语言说明复杂难懂的概念。传统的教学三部曲是:提出概念—解释概念—举例说明。我在多年教学实践中对于计算机应用课程总结了新的三部曲:提出问题—介绍解决问题的方法—归纳出必要的概念和结论。从具

体到抽象,从实际到理论,从个别到一般。这是符合人们的认识规律的。实践证明,这样做已取得了很好的效果。

为了推动高校的计算机基础教育,我在1996年主编了《计算机教育丛书》,由电子工业出版社出版。编写这套丛书的指导思想是20个字:“内容新颖、实用性强、概念清晰、通俗易懂、层次配套”(也可简单地概括为:“新颖、实用、清晰、通俗、配套”)。先后出版的近20种供大学非计算机专业使用的教材,受到高校广大师生的欢迎,几年内发行达75万册,大家认为它定位准确、程度适当、内容丰富、通俗易懂,便于自学。

在进入21世纪之际,我们根据新时期的要求,按照上面所述的指导思想,重新进行规划,对原有的教材进行了筛选,淘汰了部分内容已过时的教材,同时根据计算机技术和高校计算机基础教育的发展组织了一些新教材,并对原有教材进行了修订补充,以实现推陈出新,不断提高。

我们遴选了具有丰富教学经验的高校老师编写这套教材。在这套系列教材中,我们提供了多种课程的教材供各校选用,其中包括必修课和选修课。不同专业、不同层次的学校都可以从中选到合用的教材,我们还将根据发展不断推出新的教材。

参加本丛书策划、组织和编写工作的有:谭浩强、史济民、薛淑斌、吴功宜、边奠英、徐士良、赵鸿德、李盘林、孟宪福、张基温、宋国新、徐安东、毛汉书、李凤霞、许向荣、周晓玉、张玲、刘星、秦建中、王兴岭等。电子工业出版社对本丛书的出版给予了大力的支持,使得本丛书得以顺利出版。

由于我们的水平和经验有限,加以计算机科学技术发展很快,本丛书肯定会有不少缺点和不足,诚恳地希望专家和读者不吝指正,我们将继续努力工作,使本丛书能尽量满足读者的要求。

全国高等院校计算机基础教育研究会会长  
《21世纪计算机基础教育系列教材》主编

谭浩强

2001年7月1日

## 前 言

“数据结构”是 20 世纪 70 年代起设立的计算机专业基础课,多年来不断发展、完善,日益受到人们的重视,现在是公认的计算机科学专业的核心课程。著名的计算机学术团体,美国的 ACM 和 IEEE-CS 历次提出的课程设置报告都将其作为骨干课程或核心课程。经济信息管理计算机应用专业的学生亦应接受严格的、系统的有关计算机科学的训练,学习该课程同样具有重要的意义。

数据结构主要研究数据间的联系(数据的逻辑结构),数据在计算机中的存储方法(数据的物理结构),以及处理不同结构数据的算法。通过数据结构课程的学习,可使学生掌握用计算机进行数据处理的基本原理和方法,提高程序设计的能力。

本教材的主要对象是经济信息管理专业或者其他计算机应用专业的大学本科、专科学生,亦可供从事计算机应用工作的管理人员和技术人员学习参考。信息管理专业不同于一般的计算机软件专业,较多地偏重于基本数据结构的应用。从这一角度出发,本教材较为详尽地介绍了各种数据结构的实现和应用方法,以及各种算法在时间和空间上的分析方法,而略去了有关算法分析的烦琐的数学推导和证明。同时,压缩了有关动态存储管理的部分内容,加强了有关文件组织的比重。教材中有大量的实例和常用的算法,使读者便于借鉴和实践,亦可供在实际工作中参考和移植。为了突出数据结构的基本思想,本教材有关字符串、排序、查找的内容没有独立成章,而是根据其数据结构和算法的不同特点,分散在相应的章节中。因此本教材与一般数据结构教材相比,具有“简明易懂,应用方便,系统完整”的特点。

本教材共分 6 章。第 1 章着重介绍了数据结构课程的背景、研究对象和本教材中算法的描述和分析方法。第 2 章到第 5 章以线性结构、层次结构、网状结构为主线,由简到繁地介绍了顺序表、链表以及顺序和链接存储的栈和队,树和二叉树,图等几种基本数据结构及有关算法。第 6 章以顺序文件和随机文件为基础,较为详细地讨论了文件的索引技术以及索引文件、索引顺序文件、散列文件等常用文件的组织方法。

本教材大多数内容都给出了 PAD 图描述的算法。PAD 图具有描述算法结构清晰,可读性强,转换程序方便的特点,是十分有效的描述算法的工具。对于给出的 PAD 图描述的算法,读者很容易阅读理解和分析,一旦需要,可以方便地转换成任何一种结构化程序设计语言上机执行,例如 True BASIC, Pascal 或 C。

本课程实践性较强,因此在教学过程中,课外练习、上机操作是不可缺少的重要环节。为便于组织和实施教学以及有关人员自学,每章的最后都给出了一些精选的习题,其中上机题占据了绝大部分。该课程的上机应安排在 20 学时左右。通过上述教学环节,一方面加深理解课堂上的教学内容,另一方面提高编写和调试程序的能力。

本教材是在前任教师所编写的教材基础上,参考了人大、北大、清华等学校的有关教材,以及国内外相关文献,结合编者近几年的教学实践和体会,并针对信息专业学生的特点而编写的。本书在编写过程中得到了王利、李永昕、李宁、傅星和邱祝礼等同志的指导和帮助,在此表示衷心的感谢。由于水平所限,时间仓促,谬误之处在所难免,衷心希望得到读者特别是讲授此课程教师的批评指正。

编 者  
2002.1

# 目 录



<b>第 1 章 绪论</b> .....	(1)
1.1 数据结构概述 .....	(1)
1.1.1 学习数据结构课程的意义 .....	(1)
1.1.2 数据结构课程的特点 .....	(3)
1.2 研究对象 .....	(4)
1.3 算法的描述和分析 .....	(7)
1.3.1 算法的描述 .....	(7)
1.3.2 算法的分析 .....	(10)
本章小结 .....	(11)
习题 .....	(12)
<b>第 2 章 线性结构之一 —— 顺序表</b> .....	(13)
2.1 基本概念 .....	(13)
2.1.1 线性表 .....	(13)
2.1.2 线性表的运算 .....	(13)
2.1.3 线性表的存储 .....	(14)
2.2 顺序表 .....	(14)
2.2.1 顺序表的存储 .....	(14)
2.2.2 顺序表的运算 .....	(15)
2.2.3 顺序表的应用 .....	(18)
2.3 栈和队 .....	(25)
2.3.1 栈和队的概念 .....	(25)
2.3.2 栈和队的特点 .....	(26)
2.3.3 栈和队的存储 .....	(26)
2.3.4 栈和队的运算 .....	(26)
2.3.5 栈和队的应用 .....	(28)
本章小结 .....	(32)
习题 .....	(32)
<b>第 3 章 线性结构之二 —— 链表</b> .....	(33)
3.1 链表的存储及运算 .....	(33)
3.1.1 链表的存储 .....	(33)
3.1.2 链表的运算 .....	(34)
3.2 链接的栈和队 .....	(38)
3.2.1 链接栈和队的逻辑表示 .....	(38)
3.2.2 链接栈和队的存储 .....	(38)
3.2.3 链接栈和队的运算 .....	(39)

3.3	链表的推广 .....	(40)
3.3.1	带头结点的链表 .....	(40)
3.3.2	循环链表 .....	(41)
3.3.3	双向链表 .....	(42)
3.3.4	多重链表 .....	(43)
3.4	链表的应用 .....	(43)
3.4.1	一元多项式的表示及相加 .....	(43)
3.4.2	位组排序 .....	(44)
3.5	用数组实现链表 .....	(46)
3.5.1	链表的数组存储及运算 .....	(46)
3.5.2	存储池 .....	(47)
	本章小结 .....	(49)
	习题 .....	(49)
<b>第4章</b>	<b>层次结构——树</b> .....	<b>(50)</b>
4.1	树的概念 .....	(50)
4.2	二叉树 .....	(52)
4.2.1	二叉树的概念 .....	(52)
4.2.2	二叉树的存储 .....	(53)
4.2.3	二叉树的性质 .....	(54)
4.2.4	一般树的二叉树表示 .....	(54)
4.2.5	二叉树的运算 .....	(55)
4.2.6	二叉树的其他存储及运算 .....	(60)
4.3	树的应用 .....	(62)
4.3.1	二叉排序树 .....	(62)
4.3.2	堆排序 .....	(64)
4.3.3	哈夫曼树及运算 .....	(68)
4.3.4	决策树 .....	(72)
4.3.5	博弈树 .....	(73)
	本章小结 .....	(76)
	习题 .....	(76)
<b>第5章</b>	<b>网状结构——图</b> .....	<b>(78)</b>
5.1	图的概念 .....	(78)
5.2	图的存储 .....	(80)
5.2.1	邻接矩阵法 .....	(80)
5.2.2	邻接表法 .....	(81)
5.2.3	十字链表法 .....	(83)
5.3	图的遍历 .....	(84)
5.3.1	深度优先遍历 .....	(85)
5.3.2	广度优先遍历 .....	(86)
5.3.3	生成树 .....	(86)

5.4	最短路径 .....	(88)
5.4.1	某源点到其余各顶点的最短路径 .....	(88)
5.4.2	每对顶点间的最短路径 .....	(90)
5.5	拓扑排序 .....	(92)
5.5.1	拓扑排序的概念 .....	(92)
5.5.2	拓扑排序的算法 .....	(93)
5.6	关键路径 .....	(94)
5.6.1	关键路径的概念 .....	(94)
5.6.2	关键路径的算法 .....	(96)
	本章小结 .....	(98)
	习题 .....	(98)
<b>第6章</b>	<b>文件组织 .....</b>	<b>(99)</b>
6.1	文件的结构 .....	(99)
6.1.1	文件的逻辑结构 .....	(99)
6.1.2	文件的物理结构 .....	(100)
6.1.3	文件的组织 .....	(101)
6.2	顺序文件和随机文件 .....	(101)
6.2.1	顺序文件 .....	(101)
6.2.2	随机文件 .....	(105)
6.3	索引技术 .....	(107)
6.3.1	索引的概念 .....	(107)
6.3.2	顺序索引 .....	(107)
6.3.3	散列索引 .....	(111)
6.3.4	二叉树索引 .....	(118)
6.3.5	B 树索引 .....	(122)
6.4	索引文件 .....	(128)
6.4.1	索引文件 .....	(129)
6.4.2	索引顺序文件 .....	(130)
6.5	散列文件 .....	(135)
	本章小结 .....	(137)
	习题 .....	(137)



# 第 1 章 绪 论

当今世界已进入了信息时代。随着计算机技术的飞速发展，计算机的应用已经深入到科研和生产的各个领域，特别是在各类数据处理和信息管理工作方面发挥着巨大作用。如企业的财务、计划、生产、技术、设备、仓库管理，图书馆的图书情报检索，医院的病案管理，学校的学籍管理，金融、证券的交易管理等。开发这样的应用软件，数据结构是重要的基础和方法。本章主要介绍数据结构课的背景、研究对象和学习方法。

## 1.1 数据结构概述

在计算机发明初期，计算机主要应用于科学计算。其应用的主要特点是算法复杂，但是处理的数据量小，关系简单，一般也不需要长期保存。那时人们并不注重数据的结构和如何更好地组织它们，当然也就没有数据结构的研究和相应的课程。到了 20 世纪 60 年代，随着计算机技术的发展，计算机的应用日益广泛和深入，计算机处理的数据越来越复杂，数据量越来越大，在“计算机程序设计”、“操作系统”和“编译原理”等课程中开始涉及到了一些有关数据如何组织、如何存储的内容。特别是 1968 年，D·E·Knuth 在其名著《The Art of Computer Programming》中较为系统地讨论了各种数据结构。从那以后，国外各大学才纷纷开设“数据结构”这门课程。国内各大学的计算机专业到 20 世纪 70 年代末也开始讲授这门课。

### 1.1.1 学习数据结构课程的意义

数据结构课程多年来不断发展、完善，日益受到人们的重视，现在是公认的计算机科学专业的核心课程。著名的计算机学术团体，美国的 ACM 和 IEEE-CS 历次提出的课程设置报告，都将数据结构作为骨干课程或核心课程。经济信息管理专业等计算机应用专业的学生亦应接受严格和系统的有关计算机科学的训练，学习该课程同样具有重要的意义。

#### 1. 从事数据处理工作的需要

随着计算机技术的发展，计算机应用的推广，现在有 80% 以上的计算机应用于数据处理或信息管理领域（企业管理：计划、财务、生产、销售，情报检索，办公自动化，银行、股票、期货、保险、医院，电子商务），其应用的特点发生了本质的变化，算法简单（大多是简单的四则运算，更多的是检索），处理对象复杂（不同的数据类型，数据间存在着内在的联系，数据量剧增，需要长期保存）。这时，如何选择合理的数据结构，如何组织数据，就成为十分重要的问题，或者说是需要首先解决的问题。

#### 【例 1.1】图书馆的图书流通管理系统

图书馆的流通管理涉及到图书信息、读者信息、流通信息等。如何存储可以占用空间最少，同时又能达到查询最为方便快捷是需要解决的根本问题。

最简单的方法是不考虑数据间的相互关系，使用一个数据文件按流通信息组织和存储

数据。如下表所示：

借书号	姓名	单位	……	图书号	书名	作者	……	借书日期	还书日期	……

这样势必造成数据存储的冗余度大。例如，一个图书馆同一种书通常会购置多本，因此有关的书名、作者等图书信息都是重复的。另外，同一个读者通常允许借多本书，这时其姓名、单位等读者信息也是冗余的。这样简单组织数据，甚至可能在对数据进行处理时难以保证数据的一致性。而如果根据数据的逻辑关系进行分类，按图书信息、读者信息、流通信息分别组织和存储，则能较好地解决上述问题。

如果只是单纯地按流通次序组织数据，在按读者或按图书名检索时，都只能采用顺序检索的方法，当数据量很大时，检索效率低下。为了能够高效地进行检索，通常还需要按姓名、书名等数据的次序建立一些辅助的索引。这些都属于数据结构需要研究的内容。

### 【例 1.2】排课表系统

无论是中小学还是大学，教务部门每个学期都需要根据教学计划排课表。假设某学校有  $m$  个教师： $X_1, X_2, \dots, X_m$  和  $n$  个班级： $Y_1, Y_2, \dots, Y_n$ 。根据课程计划，教师  $X_i$  需要给班级  $Y_j$  上  $P_{ij}$  节课。那么如何在给定的教室中制定一个课时最少的完善的课表呢？显然，在某个课时，同一个教师、同一班学生或是同一个教室只能排一门课。该问题可以转换成一个图的染色问题来解决。将教师  $X_i$  和班级  $Y_j$  分别作为图中的顶点，并用  $P_{ij}$  作为边，将授课的教师和班级相连，将该图的边用尽可能少的颜色来着色即可。对于该问题的求解，除了要考虑有关的染色算法外，一个很重要的问题就是如何存储这个由多个顶点和边组成的图。不仅要存储教师和班级的信息，还需要存储教师和班级授课的关系。这些也是数据结构需要讨论的内容。

从以上实例可以看出，对于计算机在信息管理中的应用，其主要特点与科学计算不同。其算法相应简单了，但是数据类型复杂了，特别是数据间通常存在着相互依存、相互制约的关系。因此，如何将大量的、错综复杂的数据抽象、归纳成一定的逻辑结构和物理结构，用计算机进行处理，成为十分重要的研究内容。所以说，数据结构不仅是计算机专业，也是信息管理专业等从事计算机数据处理工作相关专业的重要课程。

## 2. 提高程序设计水平的需要

数据结构并不是专门研究算法的课程，但是有关不同结构数据处理的各种算法的内容贯穿了整个课程。实际上算法也是该课程的研究对象的重要方面之一，而且数据结构其他方面的研究最后都要落实到算法上。

数据结构讨论的算法往往不是针对某个具体问题的，而是针对某一类问题的，是在较为抽象的一级上进行的。这样，一方面对算法提出了更高的要求，例如通用性、坚固性等，使难度增大；另一方面也使得算法的应用范围更广，例如讨论的有关层次结构的遍历算法可以处理各种属于层次结构的数据。这样的算法设计对提高学生的程序设计水平是很好的训练。通过课程中大量的实例，将突出优良的程序设计思想、方法和风格。通过对算法的时间效率和空间效率的分析，也将进一步提高学生程序设计的能力。

## 1.1.2 数据结构课程的特点

### 1. 计算机专业的核心课程

计算机的解题过程实际上就是对输入的数据进行各种处理，最后得到运算结果的过程。要使计算机能够自动高效地完成数据处理的工作，核心的问题是要设计出好的程序。随着计算机的发展和应用领域的扩大，数据量越来越大，数据间的联系越来越复杂，因此在程序中对数据进行组织也占据了越来越大的比重。所以计算机对问题的处理可以用图 1-1 表示。

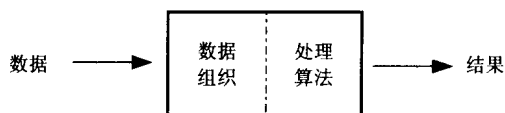


图 1-1

由此可见，计算机解题的关键是数据的组织和算法的设计，也有人将其归纳成一个数学公式（也是一本书的名称）：

$$\text{程序} = \text{算法} + \text{数据结构}$$

该公式清晰地表达出了数据结构和算法是程序设计技术的两大支柱的关系。要设计出好的程序，除了要设计好的算法之外，还必须采用合适的数据结构。所以说，数据结构是计算机专业的核心课程。而计算机应用专业研究的实质内容是利用计算机进行各种数据处理，所以数据结构课程具有同样重要的地位。

同时，数据结构课与数学、计算机硬件和计算机软件等课程都有着相当密切的关系。如图 1-2 所示。

从图中可以看出，数据结构主要研究数据类型、数据表示和数据运算，它与离散数学、应用数学相关，与计算机硬件的存储设备、体系结构相关，与计算机软件的文件组织、系统软件有关；也与数学和计算机硬件、计算机软件的交叉学科相关；但是其中关系最密切的是计算机软件。数据结构课程还是“操作系统”、“数据库原理”、“编译原理”等计算机专业课的先修课。

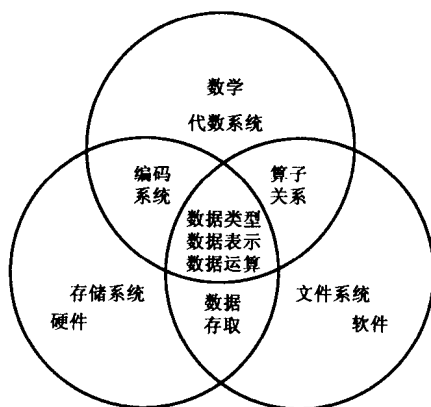


图 1-2

### 2. 专业基础课

具体地说，数据结构应划为专业基础课。它同时具备基础课和专业课两类课程的特点。作为基础课，其理论性较强，较为抽象。因此在学习数据结构课时，不能满足于了解了，知道了，而需要多问几个为什么，要举一反三。对于某个问题往往有多种算法来解，有多种数据结构可以选择。应知道哪一种更好，哪一种最好，好在什么地方。数据结构课学好了，将为计算机专业课的学习奠定良好的基础。

作为专业课，其许多内容都是直接面向应用的。因此实践环节尤其重要。只有通过大

量的练习，特别是上机练习，才有可能有较大的收获和提高，才有可能真正理解和掌握课堂和书本上的内容，才有可能学好这门课。

## 1.2 研究对象

通过图 1-2 可以看出，数据结构课主要的研究对象是各种数据表示、数据类型和数据关系的集合。其研究的范围主要包括各种数据结构的性质和特点。该课程不仅要研究数据的逻辑结构和物理结构，还要研究各种结构的数据处理算法。要了解什么是数据结构，首先要明确一些有关的概念和术语。例如什么是数据类型，什么是数据的逻辑结构，什么是数据的物理结构等。

### 1. 数据类型

计算机应用于非数值计算领域，其处理对象发生了重要的变化。不再仅仅是数值型的代数运算，还有字符型、布尔型等。而更多的是表，其处理单位往往是行，其本身也有一定的结构：每个表由若干行组成，每行由若干数据项组成，各数据项之间有一定的逻辑关系（即不允许割裂或颠倒顺序）。为了处理方便，引入了数据类型的概念。

#### (1) 基本数据类型

所谓基本数据类型，是指所应用的计算机程序设计语言或开发工具提供的数据类型。常用的有整型、实型、字符型、布尔型等。像 Pascal, C 等高级计算机程序设计语言还有指针型。而有些应用开发工具还有日期型、屏幕型等。

#### (2) 构造数据类型

所谓构造数据类型，是指所应用的计算机程序设计语言或开发工具不直接提供，而由程序设计人员根据实际问题，由基本数据类型构造的数据类型。

例如，工资单由如下数据项构成：职工号，姓名，基本工资，补贴，扣除，实发工资。如用 Pascal 语言，可以这样描述其数据类型：

```
Type
sala=Record
    no: String[5];
    name: String[8];
    bas, sub, ded, rea: Real;
End;
```

### 2. 数据元素

所谓数据元素，是指算法处理的数据的基本单位。因为本课程研究的是一般数据的处理，故数据都抽象地看做是数据元素。

#### (1) 结点

一个数据元素通常称做结点。它可能是一个数据项或是基本数据类型，也可能是由多个数据项构造的构造数据类型。而且，对于不同的算法，结点的定义是不同的。它需要根据算法的处理对象来决定。如前例中的工资单，对于计算每个人的工资的算法，其结点是一个个单个的数据项；而对于编排、打印工资单的算法，则是一行看做一个结点。

## (2) 单元

为了讨论算法方便，假设内存划分为若干存储单元，一个单元存放一个结点。显然，这里的单元也是抽象的概念。这样抽象可使我们将主要精力放在处理上，而忽略细节。

## 3. 数据逻辑结构

所谓数据的逻辑结构，是指数据间的逻辑关系。通常可用前缀、后继来描述。

### (1) 线性结构

每个结点除了头结点外都有且只有惟一的前缀，每个结点除了尾结点外都有且只有惟一的后继。具备这样逻辑关系的结点可以按其逻辑关系排成一个序列：

$$a_1, a_2, \dots, a_n$$

如果每个结点是这样的构造类型：

学号	姓名	成绩
----	----	----

且按学号大小排列，则该线性结构就是一个成绩单。

### (2) 层次结构

结点的逻辑关系可以用图 1-3 来描述。

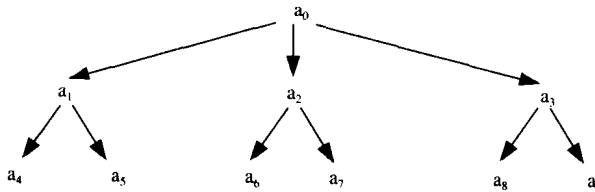


图 1-3

如果每个结点是这样的构造类型：

部门	人数	收益
----	----	----

且其值分别如下：

$a_0$ ：	农场，	195，	42300
$a_1$ ：	农业，	100，	21000
$a_2$ ：	饲养，	50，	13500
$a_3$ ：	副业，	30，	9750
$a_4$ ：	粮食，	60，	13000
$a_5$ ：	菜蔬，	35，	10000
$a_6$ ：	畜牧，	40，	8000
$a_7$ ：	渔业，	8，	6000
$a_8$ ：	食品，	12，	8000
$a_9$ ：	砖瓦，	15，	3400

这一结构虽然简单，却描述了很多信息，例如农场的部门构成。比如，农场由农业、饲养业、副业三个部门构成，而农业由粮食生产和菜蔬生产两个子部门构成……。又如各部门的非生产人数。比如，农业部门非生产人数为 5 人（共 100 人，其余所属部门人数之和为

95 人) ……，还有各部门的平均收入，非生产开支等。如果用线性结构很难甚至不可能描述这些信息。

### (3) 网状结构

结点的前缀、后继个数均没有限制。结点的逻辑关系可以用图 1-4 来描述。

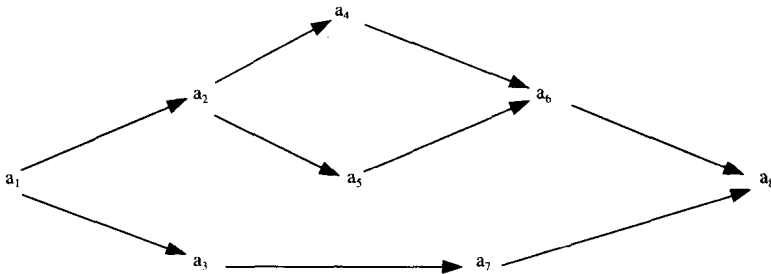


图 1-4

该结构中  $a_6$  有两个前缀。 $a_7$  从前向后是第三层，同  $a_4, a_5$  是同一层；而从后向前是第二层，同  $a_4, a_5$  又不是一层。显然这不是层次结构。该结构可以描述施工计划。假设每个结点表示一个施工项目，则该逻辑结构说明项目  $a_2$  必须在项目  $a_1$  完工后方可施工；项目  $a_6$  必须在项目  $a_4$  和项目  $a_5$  完工后方可施工；而项目  $a_7$  同项目  $a_2, a_4, a_5, a_6$  等没有制约关系，可以同时施工。

## 4. 数据物理结构

因为要研究的数据处理是在计算机上实施的，所以必须要考虑各种逻辑结构的数据在计算机中如何存储。所谓数据的物理结构，是指数据在计算机中的存储结构。它既要存储数据本身，还要能够体现出数据的逻辑关系，还需考虑算法的处理方便。

### (1) 顺序存储

所谓顺序存储，即将各结点依次顺序存储在计算机的连续单元中，通过各结点的邻接关系来体现逻辑结构——结点间的前缀、后继关系。显然，这种物理结构一般只能实现逻辑结构中的线性结构（特殊的层次结构也可用该物理结构实现）。

### (2) 链接存储

所谓链接存储，是将各结点任意存储在计算机的非连续单元，而通过“链”来描述数据的逻辑关系。即每个结点除存储数据值本身外，还用一或若干个指针，存放其后继结点的地址。

## 5. 数据处理

所谓数据的处理，是指数据的各种运算。但这里的运算是广义的，不仅包括通常意义的代数运算，更多的是非数值数据的运算。常用的算法有以下几种：

- 查找：在已存储的数据中，查找满足一定条件的结点；
- 插入：在已存储的数据中的指定位置，增加新结点；
- 删除：将指定的结点，从已存储的数据结构中删除；
- 更新：在已存储的数据结构中，修改指定的结点的值；
- 排序：将已存储的数据按某一指定的顺序重新排列。

## 6. 数据结构

数据结构可以从计算机科学、集合论等不同角度给予不同的定义。通过上述讨论，可以这样理解：

数据结构是按某种逻辑关系组织起来的一批数据，按一定的映射方式将它们存储在计算机中，并对这些数据进行各种运算。总之，数据结构的研究对象是数据的逻辑结构、数据的物理结构和数据的处理算法。其中数据的处理算法是数据的逻辑结构和物理结构的媒介，它是根据数据的逻辑结构操作，而在数据的物理结构上实施各种操作。

### 1.3 算法的描述和分析

通过前两节的介绍，可以看到数据结构主要研究数据间的联系、数据在计算机中的存储方法以及处理不同结构数据的算法这三个方面。在这三个方面中，算法占据了特别重要的地位。所谓算法，是一个有关指令的有限集合，它必须符合有穷性、确定性、有效性等准则，一般可以理解为解决某个问题的一系列的处理步骤。例如要输出 1, 2, ..., 10，其算法可描述如下：

- ① 将 1 赋予变量  $i$ ；
- ② 如果  $i > 10$  则停止处理；
- ③ 显示  $i$ ；
- ④ 将  $i+1$  赋予变量  $i$ ；
- ⑤ 转向执行②。

从以上算法描述可以看出，算法看起来很像程序，但是实际上二者具有一定的差异。程序可以不满足算法准则，例如操作系统程序可以永不终止。本课程主要讨论算法，而不讨论程序，是因为二者之间的另一个重要差别。这就是程序应该是可执行的，因而具有某种依赖性。同样的处理，不同的机器，程序是不同的；同样的处理，同样的机器，不同的程序语言，程序也是不同的；甚至同样的处理，同样的机器，同样的程序语言，由于版本不同，其程序也会有差异。而算法则没有这种依赖性，可用各种语言（包括自然语言），甚至图形来形象地描述。而且一旦需要，可以较方便地将算法转换成特定机器、特定语言的程序。

#### 1.3.1 算法的描述

算法可以使用多种方式来描述。本书主要使用下述两种方式。

##### 1. 类 Pascal 语言

本书的算法，特别是开始部分的算法通常用类 Pascal 语言描述。所谓类 Pascal 语言，即对 Pascal 程序设计语言做了一些简化处理，突出描述算法的核心部分，放弃枝节问题的描述（例如烦琐的变量说明），而且只使用一般程序设计语言都具有的语句。因此，没有学过 Pascal 语言的学生完全可以使用学过的其他任何程序设计语言，完成有关的算法，当然最好是结构化的程序设计语言。本书中所用类 Pascal 语言的主要内容如下。

所有算法都以过程和函数形式给出：

Procedure <过程名> (<参数表>);

Begin

<类 Pascal 语句>

End;

Function <过程名> (<参数表>) <类型名>;

Begin

<类 Pascal 语句>

End;

### ① 顺序结构语句

赋值语句: <变量名>:=<表达式>;

输入语句: Read (<变量表>);

输出语句: Write (<变量表>); 或 Writeln (<变量表>);

### ② 分支结构语句

条件语句: If <条件表达式> Then <语句序列 1> [Else <语句序列 2>];

多分支语句:

Case

<条件表达式 1>: <语句序列 1>;

<条件表达式 2>: <语句序列 2>;

.....

<条件表达式 n>: <语句序列 n>;

[Else <语句序列 n+1>];

### ③ 重复结构语句

For 循环语句: For <变量>:=<初值> To/Downto <终值> Do <语句序列>;

While 循环语句: While <条件表达式> Do <语句序列>;

Until 循环语句: Repeat <语句序列> Until <条件表达式>;

### ④ 过程或函数调用语句

过程调用语句: <过程名> (<参数表>);

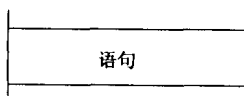
函数调用语句: <变量名>:=<函数名> (<参数表>)

## 2. PAD 图

为了使算法的思想更加清晰,便于理解,本书的绝大多数算法都给出了 PAD 图。所谓 PAD 图,即问题分析图(Problem Analysis Diagram),它是一种很好的算法描述工具,具有画法简单,层次清楚,可读性好,转换程序容易等优点。

下面是各种算法语句的 PAD 图,在书中后面的章节里,我们将直接使用这些 PAD 图,不再标注。

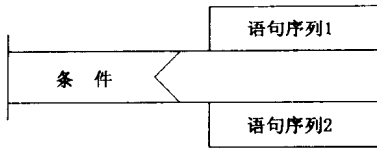
### ① 顺序结构语句



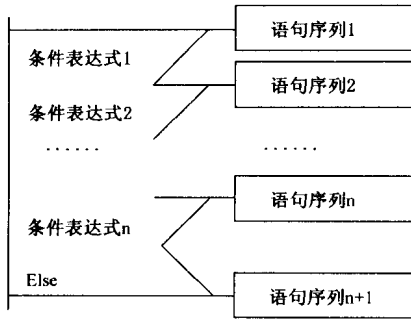


## ② 分支结构语句

### • 条件语句

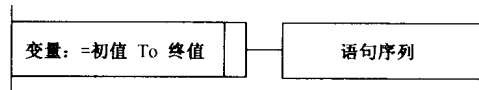


### • 多分支语句

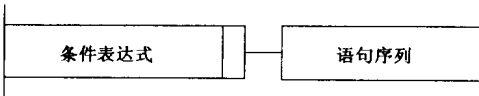


## ③ 重复结构语句

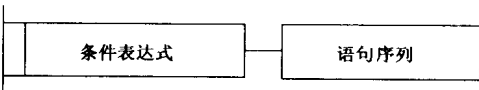
### • For 循环语句



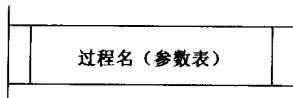
### • While 循环语句



### • Repeat 循环语句



### • 过程语句



### • 起始与终止语句

