

清华大学 计算机系列教材

殷人昆 编著

# 数据结构 (C语言描述)



清华大学出版社



清华大学 计算机系列教材

殷人昆 编著

# 数据结构 (C语言描述)

清华大学出版社  
北京

## 内 容 简 介

本书是根据2007年教育部颁发的《高等学校计算机科学与技术专业公共核心知识体系与课程》规范和2011年修订的《全国硕士研究生入学统一考试计算机科学与技术学科联考计算机学科专业基础综合考试大纲》编写的数据结构主教材。全书共分10章。第1章介绍数据结构的地位和主要知识点、数据结构和算法的基本概念和算法分析的简单方法,以及C语言编程的要点。第2章~第10章对应考试大纲的6个知识单元,包括线性表、栈、队列与数组,树与二叉树,图,查找,排序等,并做了适当延伸。作者在讨论每一个知识单元时,结合30多年教学的经验 and 考试辅导的体会,合理安排了教材内容,力求透彻、全面。对学生读书容易忽略的地方和隐藏在书中所讨论问题背后的东西都有适当的提示。

本书既可作为高校计算机科学与技术专业和软件工程专业本科生学习数据结构与算法课程的教材,也可以作为计算机专业考研的辅导教材或其他计算机及软件考试的复习教材,还可以作为从事计算机或软件系统开发的人员参考的学习资料。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

### 图书在版编目(CIP)数据

数据结构: C语言描述/殷人昆编著. —北京: 清华大学出版社, 2012. 10

清华大学计算机系列教材

ISBN 978-7-302-29119-0

I. ①数… II. ①殷… III. ①数据结构—高等学校—教材 ②C语言—程序设计—高等学校—教材  
IV. ①TP311.12 ②TP312

中国版本图书馆CIP数据核字(2012)第132038号

责任编辑: 龙启铭 战晓雷

封面设计: 常雪影

责任校对: 时翠兰

责任印制: 李红英

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦A座 邮 编: 100084

社总机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈: 010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者: 清华大学印刷厂

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 27.5

字 数: 671千字

版 次: 2012年10月第1版

印 次: 2012年10月第1次印刷

印 数: 1~3000

定 价: 43.00元

---

产品编号: 046562-01

# 序

“清华大学计算机系列教材”已经出版发行了 30 余种,包括计算机科学与技术专业的基础数学、专业技术基础和专业等课程的教材,覆盖了计算机科学与技术专业本科生和研究生的主要教学内容。这是一批至今发行数量很大并赢得广大读者赞誉的书籍,是近年来出版的大学计算机专业教材中影响比较大的一批精品。

本系列教材的作者都是我熟悉的教授与同事,他们长期在第一线担任相关课程的教学工作,是一批很受本科生和研究生欢迎的任课教师。编写高质量的计算机专业本科生(和研究生)教材,不仅需要作者具备丰富的教学经验和科研实践,还需要对相关领域科技发展前沿的正确把握和了解。正因为本系列教材的作者们具备了这些条件,才有了这批高质量优秀教材的产生。可以说,教材是他们长期辛勤工作的结晶。本系列教材出版发行以来,从其发行的数量、读者的反映、已经获得的国家级与省部级的奖励,以及在各个高等院校教学中所发挥的作用上,都可以看出本系列教材所产生的社会影响与效益。

计算机学科发展异常迅速,内容更新很快。作为教材,一方面要反映本领域基础性、普遍性的知识,保持内容的相对稳定性;另一方面,又需要跟踪科技的发展,及时地调整和更新内容。本系列教材都能按照自身的需要及时地做到这一点。如王爱英教授等编著的《计算机组成与结构》、戴梅尊教授等编著的《微型计算机技术及应用》都已经出版了第四版,严蔚敏教授的《数据结构》也出版了三版,使教材既保持了稳定性,又达到了先进性的要求。

本系列教材内容丰富,体系结构严谨,概念清晰,易学易懂,符合学生的认知规律,适合于教学与自学,深受广大读者的欢迎。系列教材中多数配有丰富的习题集、习题解答、上机及实验指导和电子教案,便于学生理论联系实际地学习相关课程。

随着我国进一步的开放,我们需要扩大国际交流,加强学习国外的先进经验。在大学教材建设上,我们也应该注意学习和引进国外的先进教材。但是,“清华大学计算机系列教材”的出版发行实践以及它所取得的效果告诉我们,在当前形势下,编写符合国情的具有自主版权的高质量教材仍具有重大意义和价值。它与国外原版教材不仅不矛盾,而且是相辅相成的。本系列教材的出版还表明,针对某一学科培养的要求,在教育部等上级部门的指导下,有计划地组织任课教师编写系列教材,还能促进对该学科科学、合理的教学体系和内容的研究。

我希望今后有更多、更好的我国优秀教材出版。

清华大学计算机系教授,中国科学院院士

张钹

# 前 言

2007年,为适应计算机科学与技术专业人才培养的需求,教育部高等学校计算机科学与技术教学指导委员会发表了《高等学校计算机科学与技术专业公共核心知识体系与课程》规范。按照该规范的规定,数据结构课程涵盖的知识单元有5个:基本算法(AL3)、算法与问题求解(PF2)、基本数据结构(PF3)、递归(PF4)和分布式算法(AL4)。

2008年,教育部决定从2009年起,对全国硕士研究生入学统一考试计算机科学与技术学科的初试科目进行调整,把数据结构课程列入了统考的考试科目之一。

本书的编写完全遵照了《全国硕士研究生入学统一考试计算机科学与技术学科联考计算机学科专业基础综合考试大纲》的要求,并参考了《高等学校计算机科学与技术专业公共核心知识体系与课程》规范。为了让学生切实领会课程知识,掌握解题方法,本书在以下几方面做了很多工作。

(1) 采用“工科”思维,启发学生掌握“化复杂为简单”的方式,从问题入手,通过“问题/子问题”分解,寻找解决方案。

(2) 完全涵盖了考试大纲的6个知识点:线性表、栈、队列与数组,树与二叉树,图,查找,排序。对基本知识点讲深讲透,通过多种应用举例,让学生了解不同问题需要采取什么方法来应对。

(3) 通过大量习题,从不同视点、不同层面训练学生。见多才能识广,才能培养出联想能力,提高分析问题和解决问题的能力。

(4) 配套辅助教材《数据结构习题精析与考研辅导》和《数据结构精讲》(即将出版)提供了多达600多道题的参考答案和解析,并就关键点进行点拨;另外,提供了多套模拟题。

本书各章除了对数据结构的基本内容做了详细讨论之外,还针对学生在与教师的互动中提出的很多疑问,以及学生在考试和做业务中暴露出来的理解上的偏差,在正文中做了正面的阐述,并通过穿插在正文中的“想想看”来提醒学生注意哪些问题是很容易被忽略的。

本书在章节安排上遵照了考试大纲的顺序。在讲解数据结构的基本知识点的过程中,引入了最常用的穷举、递推、迭代、递归等算法的设计,在后续章节中陆续引入了分治、减治、回溯、动态规划、剪枝和贪婪等算法设计策略,使得算法与数据结构的设计互相渗透,有机融合。在讨论查找、排序等算法时,特别强调了算法分析,在布置上机作业或实训中让学生通过实际比较来了解算法分析的方法。

全书采用C语言作为数据结构及算法的描述工具,适当采用C++的少量语句以简化程序。算法描述力求结构化,注重编程风格,每个算法基本保持在100行之内,可读性强。

本书可以作为计算机科学与技术专业和软件工程专业本科生的教材,也可以作为计算机专业考研(硕士研究生、工程硕士研究生和博士研究生)的复习教材,还可以供使用计算机做系统开发的人员学习使用。

因为本书是从作者多年在清华大学讲授“数据结构”课程的课件取材,并经过一定改编而成的,因此在叙述上适合学生的特点,在内容上浅显易懂,在选材上舍弃了很多超出规范

和大纲的内容,增加了许多在老教材中忽略了的内容。

各章所附习题不包括选择题,但精选了大量综合应用题,这些习题的参考解答参见作者的配套教材《数据结构习题精析与考研辅导》和《数据结构精讲》。

由于作者的水平有限,可能在某些方面有考虑不周的地方,使书中难免存在疏漏或错误,诚恳希望读者提出宝贵意见。

作者的 E-mail 地址是 [yinrk@tsinghua.edu.cn](mailto:yinrk@tsinghua.edu.cn) 或 [yinrk@sohu.com](mailto:yinrk@sohu.com)。

殷人昆

2012年9月于清华园荷清苑

# 目 录

<b>第 1 章 绪论</b> .....	1
1.1 数据结构的概念及分类 .....	1
1.1.1 为什么要学习数据结构.....	1
1.1.2 与数据结构相关的基本术语.....	2
1.1.3 数据结构的分类.....	5
1.1.4 数据结构的存储结构.....	7
1.1.5 定义在数据结构上的操作.....	8
1.1.6 “好”的数据结构.....	9
1.2 使用 C 语言描述数据结构 .....	9
1.2.1 C 语言的数据类型.....	9
1.2.2 算法的控制结构 .....	10
1.2.3 算法的函数结构 .....	11
1.2.4 动态存储分配 .....	14
1.2.5 逻辑和关系运算的约定 .....	15
1.2.6 输入与输出 .....	16
1.3 算法和算法设计.....	17
1.3.1 算法的定义和特性 .....	17
1.3.2 算法的设计步骤 .....	17
1.3.3 算法设计的基本方法 .....	18
1.4 算法分析与度量.....	21
1.4.1 算法的评价标准 .....	22
1.4.2 算法的时间和空间复杂度度量 .....	22
1.4.3 算法的渐进分析 .....	25
小结 .....	29
习题 .....	29
<b>第 2 章 线性表</b> .....	32
2.1 线性表.....	32
2.1.1 线性表的定义和特点 .....	32
2.1.2 线性表的主要操作 .....	33
2.2 顺序表.....	34
2.2.1 顺序表的定义和特点 .....	34
2.2.2 顺序表的结构定义 .....	35
2.2.3 顺序表主要操作的实现 .....	36
2.2.4 顺序表主要操作的性能分析 .....	38

2.2.5	顺序表的应用举例 .....	40
2.3	单链表 .....	41
2.3.1	单链表的定义和特点 .....	41
2.3.2	单链表的结构定义 .....	42
2.3.3	单链表中指针的操作 .....	42
2.3.4	单链表中的插入与删除 .....	43
2.3.5	带头结点的单链表 .....	46
2.3.6	单链表主要操作的性能分析 .....	48
2.3.7	单链表的顺序访问与尾递归 .....	49
2.3.8	单链表的应用举例 .....	51
2.4	顺序表与线性链表的比较 .....	53
2.5	线性链表的其他变形 .....	55
2.5.1	循环链表 .....	55
2.5.2	双向链表 .....	59
2.5.3	静态链表 .....	62
2.6	线性表的应用：一元多项式及其运算 .....	63
2.6.1	一元多项式的表示 .....	64
2.6.2	多项式的结构定义 .....	65
2.6.3	多项式的加法 .....	66
2.6.4	多项式的乘法 .....	68
小结	.....	69
习题	.....	70
<b>第3章</b>	<b>栈和队列 .....</b>	<b>74</b>
3.1	栈 .....	74
3.1.1	栈的概念 .....	74
3.1.2	顺序栈 .....	75
3.1.3	链式栈 .....	80
3.1.4	栈的混洗 .....	82
3.2	队列 .....	83
3.2.1	队列的概念 .....	83
3.2.2	循环队列 .....	85
3.2.3	链式队列 .....	88
3.3	栈的应用 .....	90
3.3.1	数制转换 .....	91
3.3.2	括号匹配 .....	91
3.3.3	表达式的计算与优先级处理 .....	93
3.3.4	栈与递归的实现 .....	98
3.4	队列的应用 .....	100
3.4.1	打印杨辉三角形与逐行处理 .....	100



3.4.2	电路布线与两点间的最短路径	102
3.5	在算法设计中使用递归	105
3.5.1	汉诺塔问题与分治法	105
3.5.2	迷宫问题与回溯法	107
3.5.3	计算组合数与动态规划	111
3.6	双端队列	113
3.6.1	双端队列的概念	113
3.6.2	输入受限的双端队列	113
3.6.3	输出受限的双端队列	114
3.6.4	双端队列的顺序存储表示	115
3.6.5	双端队列的链接存储表示	117
小结		117
习题		118
<b>第4章</b>	<b>数组、串和广义表</b>	122
4.1	数组	122
4.1.1	一维数组	122
4.1.2	多维数组	124
4.1.3	数组的应用举例	126
4.2	特殊矩阵的压缩存储	127
4.2.1	对称矩阵的压缩存储	128
4.2.2	三对角矩阵的压缩存储	129
4.3	稀疏矩阵	131
4.3.1	稀疏矩阵的概念	131
4.3.2	稀疏矩阵的三元组表表示	131
4.3.3	稀疏矩阵的十字链表表示	132
4.4	字符串	134
4.4.1	字符串的概念	134
4.4.2	字符串的初始化和赋值	134
4.4.3	C语言中有关字符串的库函数	135
4.4.4	自定义字符串的存储表示	137
4.4.5	串的模式匹配	141
4.5	广义表	147
4.5.1	广义表的概念	147
4.5.2	广义表的性质	148
4.5.3	广义表的链接表示	149
4.5.4	三元多项式的表示	151
小结		153

习题	154
<b>第 5 章 树与二叉树</b>	<b>157</b>
5.1 树的基本概念	157
5.1.1 树的定义和术语	157
5.1.2 树的基本操作	160
5.2 二叉树	161
5.2.1 二叉树的概念	161
5.2.2 二叉树的性质	161
5.2.3 二叉树的主要操作	163
5.3 二叉树的存储表示	165
5.3.1 二叉树的顺序存储表示	165
5.3.2 二叉树的链表存储表示	166
5.4 二叉树的遍历	167
5.4.1 二叉树遍历的递归算法	167
5.4.2 递归遍历算法的应用举例	168
5.4.3 二叉树遍历的非递归算法	171
5.4.4 非递归遍历算法的应用举例	176
5.4.5 二叉树的计数	178
5.5 线索二叉树	181
5.5.1 线索二叉树的概念	182
5.5.2 线索二叉树的种类	182
5.5.3 中序线索二叉树的建立和遍历	183
5.5.4 前序与后序线索二叉树	185
5.6 树与森林	187
5.6.1 树的存储表示	187
5.6.2 森林与二叉树的转换	190
5.6.3 树与森林的深度优先遍历	192
5.6.4 树与森林的广度优先遍历	194
5.6.5 树遍历算法的应用举例	195
小结	196
习题	197
<b>第 6 章 树与二叉树的应用</b>	<b>201</b>
6.1 二叉查找树	201
6.1.1 二叉查找树的概念	201
6.1.2 二叉查找树的查找	202
6.1.3 二叉查找树的插入	203
6.1.4 二叉查找树的删除	205
6.1.5 二叉查找树的性能分析	207
6.2 AVL 树	209

6.2.1	AVL 树的概念 .....	209
6.2.2	平衡化旋转 .....	210
6.2.3	AVL 树的插入 .....	212
6.2.4	AVL 树的删除 .....	214
6.2.5	AVL 树的性能分析 .....	216
6.3	Huffman 树 .....	218
6.3.1	带权路径长度的概念 .....	218
6.3.2	Huffman 树的概念 .....	219
6.3.3	最优判定树 .....	222
6.3.4	Huffman 编码 .....	224
6.4	堆 .....	225
6.4.1	小根堆和大根堆 .....	225
6.4.2	堆的建立 .....	227
6.4.3	堆的插入 .....	228
6.4.4	堆的删除 .....	230
6.5	并查集 .....	230
6.5.1	并查集的定义及其实现 .....	230
6.5.2	并查集操作的分析和改进 .....	232
6.6	八皇后问题与树的剪枝 .....	234
6.6.1	八皇后问题的提出 .....	234
6.6.2	八皇后问题的状态树 .....	235
6.6.3	八皇后问题算法 .....	237
	小结 .....	238
	习题 .....	238
<b>第 7 章</b>	<b>图 .....</b>	<b>242</b>
7.1	图的基本概念 .....	242
7.1.1	与图有关的若干概念 .....	242
7.1.2	图的基本操作 .....	245
7.2	图的存储结构 .....	247
7.2.1	图的邻接矩阵表示 .....	247
7.2.2	图的邻接表表示 .....	251
7.2.3	邻接矩阵表示与邻接表表示的比较 .....	256
7.2.4	图的邻接多重表表示 .....	256
7.3	图的遍历 .....	258
7.3.1	深度优先搜索 .....	258
7.3.2	广度优先搜索 .....	260
7.3.3	连通分量 .....	261
7.3.4	重连通图 .....	263
7.3.5	有向图的强连通分量 .....	265

7.4	最小生成树 .....	267
7.4.1	最小生成树求解和贪婪法 .....	267
7.4.2	Kruskal 算法 .....	269
7.4.3	Prim 算法 .....	272
7.5	最短路径 .....	274
7.5.1	非负权重的单源最短路径 .....	274
7.5.2	所有顶点之间的最短路径 .....	278
7.5.3	无权重的最短路径 .....	281
7.6	活动网络 .....	282
7.6.1	AOV 网络与拓扑排序 .....	282
7.6.2	AOE 网络与关键路径法 .....	286
	小结 .....	291
	习题 .....	291
<b>第 8 章</b>	<b>查找</b> .....	<b>297</b>
8.1	查找的基本概念 .....	297
8.1.1	查找的概念 .....	297
8.1.2	查找算法的性能分析 .....	298
8.2	顺序查找法 .....	298
8.2.1	在顺序表上的顺序查找算法 .....	298
8.2.2	在线性链表上的顺序查找算法 .....	302
8.3	折半查找法 .....	302
8.3.1	折半查找法 .....	302
8.3.2	次优查找树: 折半查找的改进方法 .....	305
8.3.3	斐波那契查找: 折半查找的变形 .....	310
8.3.4	插值查找: 折半查找的变形 .....	312
8.4	B 树 .....	313
8.4.1	索引顺序表与分块查找 .....	313
8.4.2	多级索引结构与 $m$ 叉查找树 .....	315
8.4.3	B 树的概念 .....	316
8.4.4	B 树上的查找 .....	317
8.4.5	B 树上的插入 .....	318
8.4.6	B 树上的删除 .....	320
8.4.7	$B^+$ 树 .....	321
8.5	散列表及其查找 .....	324
8.5.1	散列的概念 .....	324
8.5.2	常见的散列函数 .....	325
8.5.3	解决冲突的开地址法 .....	328
8.5.4	解决冲突的链地址法 .....	335
8.5.5	散列法分析 .....	337

小结	338
习题	339
<b>第 9 章 内排序</b>	<b>344</b>
9.1 排序的概念	344
9.1.1 排序的概念	344
9.1.2 排序算法的性能	345
9.1.3 数据表的结构定义	346
9.2 插入排序	347
9.2.1 直接插入排序	347
9.2.2 基于链表的直接插入排序	349
9.2.3 折半插入排序	350
9.2.4 希尔排序	352
9.3 交换排序	353
9.3.1 起泡排序	354
9.3.2 快速排序	356
9.3.3 快速排序的改进算法	359
9.4 选择排序	360
9.4.1 简单选择排序	361
9.4.2 堆排序	362
9.4.3 锦标赛排序	366
9.5 归并排序	367
9.5.1 二路归并排序的设计思路	367
9.5.2 二路归并排序的递归算法	367
9.5.3 基于链表的归并排序算法	369
9.5.4 迭代的归并排序算法	371
9.6 基数排序	373
9.6.1 基数排序	373
9.6.2 MSD 基数排序	374
9.6.3 LSD 基数排序	376
9.7 内排序算法的分析和比较	379
9.7.1 排序方法的下界	379
9.7.2 分布计数排序	380
9.7.3 各种内排序方法的比较	381
小结	384
习题	385
<b>第 10 章 外排序</b>	<b>389</b>
10.1 主存储器 and 外存储器	389
10.1.1 磁带	389
10.1.2 磁盘存储器	390

10.1.3	缓冲区	391
10.2	基于磁盘的外排序过程	392
10.2.1	基于磁盘排序的过程	392
10.2.2	基于磁盘排序的性能分析	393
10.3	$m$ 路平衡归并	394
10.3.1	$m$ 路平衡归并的过程	394
10.3.2	用败者树选最小记录	395
10.3.3	败者树的构造	396
10.4	初始归并段的生成	398
10.4.1	置换-选择排序	398
10.4.2	用败者树实现置换-选择排序	398
10.4.3	用小根堆实现置换-选择排序	399
10.4.4	置换-选择排序的性能分析	402
10.5	最佳归并树	403
10.6	并行操作的缓冲区处理	405
10.6.1	使用固定缓冲区实现并行操作	405
10.6.2	使用缓冲区队列实现并行操作	405
10.7	磁带归并排序	407
10.7.1	平衡归并排序	408
10.7.2	多步归并排序	408
10.7.3	多步归并排序初始归并段的分配	409
	小结	410
	习题	411
	<b>附录 A 程序索引</b>	<b>413</b>
	<b>附录 B 实训作业要求与样例</b>	<b>420</b>
	<b>参考文献</b>	<b>426</b>

# 第 1 章 绪 论

开发一个计算机系统,最基本的工作就是编程。没有程序,没有数据,再好的计算机硬件也是没有用的。譬如修建高速铁路,没有好的运行计划程序、调度程序、网络通信程序和机车控制程序,不可能让整个线路运转起来。因此,一个计算机系统离不开程序和数据。而算法和数据结构是程序的核心,是高效解决问题方法的选择以及为支持问题解决所采取的数据组织方式。

当前,算法和数据结构已成为计算机科学与技术学科和软件工程学科的一门重要的专业基础课程,也是许多后续课程(如操作系统、计算机系统结构、计算机网络、编译原理、数据库、人工智能等)的先修课程。它不仅是计算机和软件工程专业专业的必修课,也是许多非计算机专业学生的选修课和从事系统开发人员的培训课程。

## 1.1 数据结构的概念及分类

### 1.1.1 为什么要学习数据结构

人们在使用计算机解决一个个具体问题时,通常遵循的步骤是“理解需求—分析建模—设计解决方案—编程实现”。理解需求好比在考试中面对一个问题,首先要弄懂这个问题到底要你干什么。分析建模就是把你问题的理解按不同视角进行整理,用模型进行抽象。设计解决方案将根据模型提出求解算法,并基于算法设计适当的数据结构和程序框架。编程实现将把程序框架用某种程序设计语言翻译成计算机可执行的程序,经过测试和改错,最终得到适用的程序。

**【例 1-1】** 开发大学选课系统。假设某学期为计算机系的 160 名本科生设置了 40 门可选课程,限定每位学生一个学期只能选 6 门课程。这样,学生和课程之间出现了多对多的关系,如图 1-1(a)所示。如果引入一个交互实体“选课”,学生和课程之间的关系就转化为两个一对多的关系,如图 1-1(b)所示。

为解决以上问题,首先应理解需求,每学期开学前,为计算机系的学生设置可选课程的列表,学生在开学的前两周可以选课和退选。每位学生限定最多一学期选 3 门课,每门课程限定最多可接受 40 名学生。第二步,分析建模。处理模型包括选课的流程和退选的流程,数据模型包括“学生”、“课程”和“选课”实体的构成和它们之间关系的描述。第三步,设计解决方案,包括设计各个数据实体的数据结构和存储映像、相应操作的实现逻辑以及用户界面的显示和交互的流程。第四步,用 C 语言或 Java 语言实现程序代码,用网页制作工具实现菜单树以及数据的输入、输出和窗口的控制和切换。

这类问题属于典型的非数值型事务处理问题。类似的问题包括商店销售事务处理问题、电信计费和收费处理问题、电梯控制处理问题和银行排队问题等。

由以上事例可知,问题的最终解决取决于程序,而程序的质量又取决于算法的选择和数

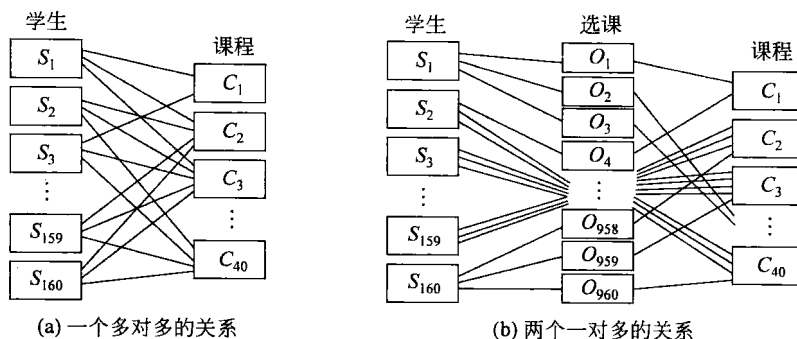


图 1-1 课程与学生之间的两种联系

据的组织方式。著名的瑞士科学家 Niklaus Wirth(沃思)在其经典著作“Algorithms + Data Structures = Programs”中强调,“程序的构成与数据结构是两个不可分割地联系在一起的问题。”他又引用 C. A. R. Hoare(霍尔)在“Notes on Data Structuring”中的名言:“不了解施加于数据上的算法就无法决定如何构造数据,反之,算法的结构和选择却常常在很大程度上依赖于作为基础的数据结构。”从而给出一个权威性的定义:程序就是在数据的某些特定的表示方式和结构的基础上对抽象算法的具体表述。

因此,学习数据结构的意义在于编写高质量的程序。因为人们的直观概念总是数据先于算法——你总得先有对象才有施加于它的算法。

## 1.1.2 与数据结构相关的基本术语

### 1. 数据(data)

我们在日常生活中会遇到各种信息,如用语言交流的思想、在战争中用于传递命令的旗语等。这些信息必须转换成数据才能在计算机中进行处理。因此,数据的定义是:数据是信息在计算机程序中的表示形式或编码形式,是描述客观事物的数、字符以及所有能输入到计算机中并被计算机程序识别和处理的符号的集合。

数据大致可分为两类,一类是数值性数据,包括整数、浮点数、复数和双精度数等,主要用于工程和科学计算以及商业事务处理;另一类是非数值数据,主要包括字符和字符串,以及文字、图形、图像和语音等。

### 2. 数据元素(data element)

数据的基本单位是数据元素,它是计算机处理或访问的基本单位。例如,一个考生名册中的每个学生记录、一个字符串中的每一个字符、一个数组的每一个数组成分都是数据元素。不同场合下数据元素可以有别名,如元素、记录、结点或表项等。

### 3. 数据项(data item)

数据元素可以是单个元素(称为原子),也可以由数据项构成。数据项又称为属性、字段或域。数据元素中的数据项可以分为两种,一种叫做初等项,如学生的性别、籍贯等,这些数据项是在数据处理时不能再分割的最小单位;另一种叫做组合项,如学生的成绩,它可以再划分为物理、化学等更小的项。

### 4. 数据结构(data structure)

在数据处理中所涉及的数据元素之间都不是孤立的,在它们之间存在着某种关系,这种



数据元素之间的关系称为结构。例如,招生考试时把所有考生按考试成绩从高到低排队,所有考生记录都将处在一种有序的序列中;又例如,在  $n$  个网站之间建立通信网络,要求以最小的代价将  $n$  个网站连通,如图 1-2(a)所示,这样,在所有网站之间形成一种树状关系;反之,要求网络中任一网站出现故障,整个网络仍然保持畅通,这样,在所有网站之间形成一种网状关系,如图 1-2(b)所示。

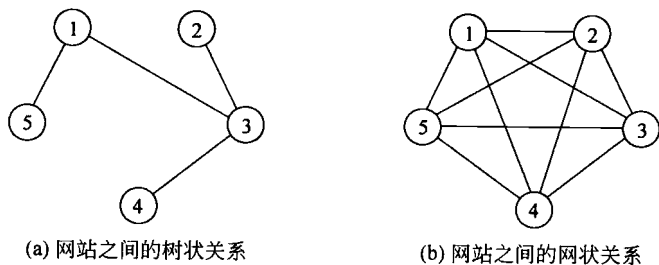


图 1-2  $n$  个网站之间的连通关系

由此可以引出数据结构的定义:数据结构是由与特定问题相关的某一数据元素的集合和该集合中数据元素之间的关系组成的。

数据结构可分为静态数据结构(static data structure)和动态数据结构(dynamic data structure)。例如,数组是最简单的静态数据结构,它的元素的个数和元素间的关系是不变的;二叉查找树和堆是动态数据结构,它们的元素个数和元素间的关系将会因元素的插入或删除而变化。

### 5. 数据对象(data object)

从狭义的观点把数据对象定义为具有一定关系的相同性质的数据元素的集合,从广义的观点把数据对象定义为一组值的集合,且数据对象应是数据抽象和处理抽象的封装体,即数据对象的声明中不但要包含属性,还要包含可用的操作。

### 6. 数据类型(data type)

从程序设计角度来看,数据类型与数据结构的概念是相通的,主要用于刻画程序中操作对象的特性。数据类型明确地或隐含地规定了在程序执行期间变量、表达式或函数所有可能取值的范围,以及作用在这些取值上的操作。因此,数据类型是一个值的集合和定义在这个值集合上的一组操作的总称。例如,C或C++语言规定了一些内置的数据类型,如整数、浮点数、字符、指针和枚举量等,它们的表示、取值和操作如表 1-1 所示。

表 1-1 C/C++ 语言中的内置数据类型

表示	char	int	unsigned int	long int	unsigned long int	float	double
类型	字符型	整型	无符号整型	长整型	无符号长整型	浮点型	双精度型
位数	8	16	16	32	32	32	64
取值	-127~127	-32 767~32 767	0~65 535	-2 147 483 647~2 147 483 647	0~4 294 967 295	6 位有效数字	10 位有效数字
操作		+, -, ×, /, %	+, -, ×, /, %	+, -, ×, /, %	+, -, ×, /, %	+, -, ×, /	+, -, ×, /