

高等学校21世纪教材

GAODENG XUOXIAO 21 SHIJI JIAOCAI

# 数据结构

## — C++语言 描述

● 陈慧南 编著



人民邮电出版社  
POSTS & TELECOM PRESS

高等学校 21 世纪教材

# 数据结构——C++语言描述

陈慧南 编著

人民邮电出版社

## 图书在版编目 (CIP) 数据

数据结构——C++语言描述/陈慧南编著. —北京: 人民邮电出版社, 2005.3

高等学校 21 世纪教材

ISBN 7-115-12941-X

I. 数... II. 陈... 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. TP311.12②TP312

中国版本图书馆 CIP 数据核字 (2004) 第 140086 号

### 内 容 提 要

本书采用面向对象的观点讨论数据结构, 并使用 C++ 语言描述。书中不仅系统地介绍各种传统的数据结构和搜索、内外排序算法, 还引入一些高级数据结构, 如伸展树和跳表。本书重视算法的时间和空间分析, 包括搜索和排序时间的下界分析。

全书条理清晰, 内容详实, 既注重数据结构和算法原理, 又十分强调程序设计训练。书中算法都有完整的 C++ 程序, 程序结构清晰, 构思精巧。所有程序都已在 VC++ 环境下编译通过并能正确运行, 它们既是学习数据结构和算法的示例, 也是很好的 C++ 程序设计示例。本书深入浅出, 配有大量的实例和图示, 并有丰富的习题, 适于自学。

本书可作为高等院校计算机科学与技术专业和其他相关专业的教材, 也可供计算机工作者和其他希望学习数据结构和算法知识的人员参考。

高等学校 21 世纪教材

### 数据结构——C++语言描述

- 
- ◆ 编 著 陈慧南  
责任编辑 滑 玉
  - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
邮编 100061 电子函件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
读者热线 010-67170985  
北京隆昌伟业印刷有限公司印刷  
新华书店总店北京发行所经销
  - ◆ 开本: 787×1092 1/16  
印张: 23.25  
字数: 552 千字 2005 年 3 月第 1 版  
印数: 1-5 000 册 2005 年 3 月北京第 1 次印刷

ISBN 7-115-12941-X/TP · 4358

定价: 29.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

## 丛书前言

当今世界，科学技术突飞猛进，知识经济已见端倪，国际竞争日趋激烈。教育在综合国力的形成中处于基础地位，国力的强弱将越来越取决于劳动者的素质，取决于各类人才的质量和数量，这对于培养和造就我国 21 世纪的一代新人提出了更加迫切的要求。21 世纪初，我国高等教育呈快速发展的势头。教材是体现教学内容和教学方法的知识载体，是进行教学的基本工具，也是深化教育教学改革、全面推进素质教育、培养创新人才的重要保证。因此，高等教育教材建设必须有一个与之相适应的快速发展。

随着计算机软硬件的不断升级换代，计算机教学内容也随之更新，尤其随着教育部“高等教育面向 21 世纪教育内容与课程体系改革”计划的实施，对教材也提出了新的要求。为此我们聘请了国内高校计算机教学方面知名的专家教授，精心策划编写了这套“高等学校 21 世纪教材”。

为真正实施精品战略，组织编写好这套教材，我们在国内高校做了系统、详细的调查，对教育部制订的教育计划做了认真的研究，还对国内外已出版的教材做了理性的分析，确立了依托国家教育计划、传播先进教学理念、为培养符合社会需要的高素质创新型人才服务的宗旨。

在本套教材的策划过程中，我们多次组织了由专家及高校一线教师参加的研讨会，对现有比较出色的教材的特点及优点进行了分析，博采众长，力求实现教材权威性与实用性的完美结合。

本套教材有如下特点：

1. 考虑到全国普通高等院校学生的知识、能力、素质的特点和实际教学情况，在编写教材时把重点放在基本理论、基础知识、基本技能与方法上。
2. 紧密结合当前技术的新发展，在阐述理论知识的同时侧重实用性。
3. 力求在概念和原理的讲述上严格、准确、精练，理论适中，实例丰富，写作风格上深入浅出，图文并茂，便于学生学习。
4. 为适应当前高校课程种类多、课时数要压缩的教学特点，教材不仅篇幅有很大的压缩，而且均配有电子教案，以满足现代教学新特点的需要，做到易教易学。
5. 所选作者均是国内有丰富教学实践经验的知名专家、教授，所编教材具有较高的权威性。

教育的改革将不会停止，教材也将会不断推陈出新。目前本套教材即将推出，将接受广大教学第一线教师的检验。

由于我们的水平和经验有限，这批教材在编审、出版工作中还存在不少缺点和不足，希望使用本套教材的学校师生和广大读者提出批评和建议，以便改进我们的工作，使教材质量不断提高。

# 编者的话

计算学科教学计划 2001 (简称 CC2001) 将计算机学科分成 14 个领域, 每个领域分成若干知识单元, 每个知识单元又包括若干主题。

数据结构知识主要属于程序设计基础 (PF)、算法与复杂性 (AL) 和程序设计语言 (PL) 等领域。掌握这些领域的知识对于利用计算机资源, 开发高效的计算机程序是非常必要的。CC2001 将这些领域的大多数知识单元规定为计算机及相关学科的本科学生必须掌握的核心知识单元。主要包括 PF3, PF4, PL4, PL6, PL9, AL1 和 AL3 等知识单元, 其中, PF3 为基本数据结构, 包括数组、字符串、堆栈、队列、树、图和散列表等主题; PF4 涉及递归技术; PL4 和 PL9 分别为数据类型和数据抽象; PL6 为面向对象程序设计; AL1 为算法分析的基本方法; AL3 是基本的计算算法, 包括查找、排序、哈希表算法、搜索树及图算法等主题。本书的内容涵盖上述知识单元。

“数据结构”是计算机科学与技术专业的一门核心课程, 也是计算机软件和应用工作者必备的专业基础。掌握扎实的数据结构知识有助于理工科学生进一步学习计算机技术, 适应更广泛的职业挑战。目前, 我国计算机类专业和大多数理工科类专业都开设课程讲授这方面的知识。

本书在内容上注意兼顾广度和深度, 不仅系统地介绍了各种传统的数据结构和各种搜索、内外排序方法, 还引入了一些比较高级的数据结构, 如伸展树和跳表。本书重视算法的时间和空间分析, 对许多算法都作了详细的性能分析, 包括搜索和排序时间的下界分析。

近年来, “数据结构”课程的内容和讲授体系有了很大的变革。该课程在介绍数据结构知识的同时, 普遍重视软件设计原理和程序设计方面的训练, 如面向对象方法和 C++ 程序设计。本书采用抽象数据类型和面向对象方法研究数据结构, 并使用 C++ 语言描述。读者通过学习, 掌握数据抽象原理和面向对象软件设计方法。

本书既注重数据结构和算法原理, 又十分强调程序设计训练。书中算法都有完整 C++ 程序, 程序结构清晰, 构思精巧。所有程序都在 VC++ 环境下编译通过并能正确运行。它们既是学习数据结构和算法的示例, 也是很好的 C++ 程序设计示例。

全书条理清晰, 内容详实, 深入浅出。书中对算法做了较详细的解释, 尽可能做到可读易懂, 并配有大量的实例和图示, 有利于读者理解算法的实质和编程思想。每章引言和结尾处的小结帮助读者了解一章的要点, 并有丰富的习题, 适于自学。读者可以通过南京邮电学院主页 [www.njupt.edu.cn](http://www.njupt.edu.cn), 点击“教育网站”栏目, 链接至作者主持的数据结构与算法的 Web 站点, 了解该课程教学和学习的有关信息。

本书是作者根据多年在南京邮电学院讲授《数据结构》和《算法设计与分析》课程的教学经验, 在由作者编写的, 使用 Pascal、C 和 C++ 描述的几本《数据结构》教材的基础上, 参考了近年来国内外出版的多种数据结构和算法的优秀教材编写而成。

全书共分 14 章。第 1 章是基础知识, 介绍什么是数据结构, 什么是算法; 介绍数据抽象和抽象数据类型的概念; 回顾面向对象方法以及 C++ 程序设计的基本特征; 给出了书中描述数据结构和算法的方法; 本章最后介绍算法分析的基本方法。

第 2 章重点讨论数据结构的两种最常用的存储表示方式: 顺序表示和链接表示。当使用 C++

语言实现时,我们往往使用数组实现顺序存储,而使用链表实现链接存储。这里,数组和链表被作为抽象数据类型的实现形式,它们几乎是实现所有 ADT 的基础。由于数组自身也可以看成一个抽象数据类型,数组作为一种抽象数据类型在第 4 章讨论。

第 3, 5, 6 章介绍几种线性数据结构:堆栈、队列、线性表和字符串。多维数组和广义表是推广的线性结构,也在这一部分讨论。

第 4 章专门讨论递归,讨论递归的概念、递归算法和递归算法的效率;介绍编译程序实现函数嵌套调用和递归函数的机制;讨论分析递归算法常用的递推关系式及其求解方法。

第 7 章和第 12 章讨论两种重要的非线性数据结构——树和图。

第 8, 9, 10 和 11 章介绍集合的表示和运算。重点讨论动态集的搜索、插入和删除运算的算法。第 8 章讨论使用线性表表示集合时,在表中搜索元素的方法:顺序搜索和二分搜索。第 9 章介绍动态集的二叉搜索树表示,包括二叉搜索树、二叉平衡树和伸展树。第 10 章使用多叉树: B 树和 Trie 树,来表示动态集, Trie 树是一种数字搜索树。第 11 章讨论跳表和散列表,它们利用随机性,获得较好的平均时间复杂度,是两种有效的表示动态集的数据结构。

第 13 章介绍多种内排序算法,并对它们作性能分析。有的算法还给出了顺序表和链表两种实现。

第 14 章讨论外排序过程和算法。文件作为一种外存的数据结构,也在第 14 章介绍。

本书可作为高等院校计算机科学与技术专业和其他相关专业的“数据结构”课程 84 学时的教学内容。对于学时数少于 84 学时的授课计划,可根据实际学时加以剪裁。作者已在目录中对难度较大,或非基本的章节标上\*号,供读者选取时参考。除带\*号内容以外的基本部分适合 48~56 学时的授课计划。

本书的编写得到了南京邮电学院和计算机科学与技术系的领导推荐和关心,在此表示衷心感谢。

书中若有不当之处,敬请读者批评指正。

编者

2004 年 6 月 28 日于南京

# 目 录

第 1 章 基础知识 .....	1
1.1 算法与数据结构 .....	1
1.2 什么是数据结构 .....	2
1.2.1 基本概念 .....	2
1.2.2 数据的逻辑结构 .....	3
1.2.3 数据的存储表示 .....	4
1.2.4 数据结构的运算 .....	4
1.3 数据抽象和抽象数据类型 .....	5
1.3.1 抽象、数据抽象和过程抽象 .....	5
1.3.2 封装与信息隐蔽 .....	6
1.3.3 数据类型和抽象数据类型 .....	6
1.3.4 数据结构与抽象数据类型 .....	7
1.4 面向对象方法 .....	7
1.4.1 面向对象方法的由来 .....	7
1.4.2 面向对象方法的基本思想 .....	7
1.4.3 面向对象方法的要素 .....	8
1.4.4 面向对象方法和抽象数据类型 .....	9
1.5 *C++程序设计概要 .....	9
1.5.1 函数与参数传递 .....	9
1.5.2 动态存储分配 .....	12
1.5.3 类与对象 .....	13
1.5.4 函数和运算符重载 .....	14
1.5.5 继承性和派生类 .....	15
1.5.6 多态性、虚函数和动态联编 .....	16
1.5.7 纯虚函数和抽象类 .....	17
1.5.8 友元函数和友元类 .....	18
1.5.9 模板函数和模板类 .....	18
1.6 描述数据结构和算法 .....	21
1.6.1 数据结构的规范 .....	21
1.6.2 实现数据结构 .....	22
1.7 算法和算法分析 .....	23
1.7.1 算法及其性能标准 .....	23
1.7.2 算法的时间复杂度 .....	24
1.7.3 使用程序步分析算法 .....	26

1.7.4	渐近表示法 .....	28
1.7.5	算法按时间复杂度分类 .....	29
1.7.6	算法的空间复杂度 .....	30
	本章小结 .....	30
	习题 .....	31
<b>第2章</b>	<b>数组和链表 .....</b>	<b>33</b>
2.1	结构和类 .....	33
2.1.1	结构 .....	33
2.1.2	结构表示元素 .....	34
2.2	指针和动态存储分配 .....	36
2.2.1	指针 .....	36
2.2.2	动态存储分配 .....	37
2.2.3	静态变量和动态变量 .....	38
2.3	数组 .....	38
2.3.1	一维数组 .....	38
2.3.2	二维数组 .....	39
2.3.3	多维数组 .....	40
2.3.4	数组和指针 .....	40
2.4	链表 .....	41
2.4.1	指向结构的指针 .....	42
2.4.2	单链表 .....	42
2.4.3	带表头结点的单链表 .....	45
2.4.4	单循环链表 .....	45
2.4.5	双向链表 .....	46
2.5	采用模拟指针的链表 .....	47
2.5.1	结点结构 .....	47
2.5.2	可用空间表 .....	48
2.6	异常处理 .....	50
	本章小结 .....	51
	习题 .....	51
<b>第3章</b>	<b>堆栈和队列 .....</b>	<b>53</b>
3.1	堆栈 .....	53
3.1.1	堆栈 ADT .....	53
3.1.2	堆栈的顺序表示 .....	54
3.1.3	堆栈的链接表示 .....	57
3.2	队列 .....	59
3.2.1	队列 ADT .....	59
3.2.2	队列的顺序表示 .....	61
3.2.3	队列的链接表示 .....	64
3.3	*表达式计算 .....	64
3.3.1	表达式 .....	64



3.3.2	中缀表达式转换为后缀表达式 .....	65
3.3.3	计算后缀表达式的值 .....	68
3.4	*演示与测试 .....	71
	本章小结 .....	75
	习题 .....	75
<b>第4章</b>	<b>*递归 .....</b>	<b>77</b>
4.1	递归和递归算法 .....	77
4.1.1	递归的概念 .....	77
4.1.2	递归算法示例 .....	78
4.1.3	递推关系 .....	80
4.2	实现递归 .....	81
4.2.1	函数调用和系统栈 .....	81
4.2.2	递归函数的性能 .....	82
4.2.3	尾递归 .....	83
4.2.4	消去递归 .....	83
	本章小结 .....	87
	习题 .....	87
<b>第5章</b>	<b>线性表和数组 ADT .....</b>	<b>88</b>
5.1	线性表 .....	88
5.1.1	线性表 ADT .....	88
5.1.2	线性表的顺序表示 .....	90
5.1.3	线性表的链接表示 .....	93
5.1.4	两种存储表示的比较 .....	96
5.2	*一元多项式算术运算 .....	97
5.2.1	多项式 ADT .....	97
5.2.2	多项式的链接表示 .....	98
5.2.3	项结点类 .....	98
5.2.4	多项式类 .....	99
5.2.5	多项式的输入和输出 .....	100
5.2.6	多项式相加 .....	101
5.2.7	多项式相乘 .....	102
5.2.8	重载运算符函数 .....	103
5.3	数组作为抽象数据类型 .....	104
5.3.1	数组 ADT .....	104
5.3.2	一维数组的 C++类 .....	105
5.4	特殊矩阵 .....	106
5.4.1	对称矩阵 .....	106
5.4.2	*带状矩阵 .....	107
5.5	稀疏矩阵 .....	108
5.5.1	稀疏矩阵 ADT .....	108
5.5.2	稀疏矩阵的三元组表 .....	109

5.5.3	稀疏矩阵转置 .....	110
5.5.4	*稀疏矩阵相乘 .....	112
5.6	*稀疏矩阵的正交链表 .....	115
5.6.1	稀疏矩阵的正交链表表示 .....	115
5.6.2	正交链表结点类 .....	116
5.6.3	正交链表类 .....	117
5.6.4	建立正交链表 .....	118
5.6.5	打印正交链表 .....	119
本章小结	.....	120
习题	.....	120
<b>第6章</b>	<b>字符串和广义表 .....</b>	<b>122</b>
6.1	字符串 .....	122
6.1.1	字符串 ADT .....	122
6.1.2	字符串的存储表示 .....	123
6.1.3	串运算的实现 .....	124
6.1.4	简单模式匹配算法 .....	125
6.1.5	*模式匹配的 KMP 算法 .....	127
6.2	*广义表 .....	131
6.2.1	广义表的概念 .....	131
6.2.2	广义表 ADT .....	132
6.2.3	广义表的存储表示 .....	133
6.2.4	广义表算法 .....	134
本章小结	.....	135
习题	.....	135
<b>第7章</b>	<b>树 .....</b>	<b>137</b>
7.1	树的基本概念 .....	137
7.1.1	树的定义 .....	137
7.1.2	基本术语 .....	138
7.2	二叉树 .....	139
7.2.1	二叉树的定义 .....	139
7.2.2	二叉树的性质 .....	140
7.2.3	二叉树 ADT .....	141
7.2.4	二叉树的存储表示 .....	142
7.2.5	二叉树类 .....	143
7.2.6	二叉树基本运算 .....	144
7.3	二叉树的遍历 .....	146
7.3.1	二叉树遍历算法 .....	146
7.3.2	二叉树遍历的递归算法 .....	147
7.3.3	二叉树遍历的应用示例 .....	149
7.4	*二叉树遍历的非递归算法 .....	151
7.4.1	遍历器类 .....	151

7.4.2	中序遍历器类 .....	153
7.4.3	后序遍历器类 .....	154
7.5	*二叉线索树 .....	157
7.5.1	二叉线索树的定义 .....	157
7.5.2	构造中序线索树 .....	158
7.5.3	遍历二叉线索树 .....	159
7.6	树和森林 .....	161
7.6.1	森林与二叉树的转换 .....	161
7.6.2	树和森林的存储表示 .....	162
7.6.3	树和森林的遍历 .....	164
7.7	*堆和优先权队列 .....	164
7.7.1	堆 .....	165
7.7.2	优先权队列 ADT .....	167
7.7.3	优先权队列类 .....	168
7.7.4	实现优先权队列 .....	168
7.8	哈夫曼树和哈夫曼编码 .....	170
7.8.1	树的路径长度 .....	171
7.8.2	哈夫曼树和哈夫曼算法 .....	172
7.8.3	哈夫曼树类 .....	173
7.8.4	构造哈夫曼树 .....	173
7.8.5	哈夫曼编码 .....	175
7.8.6	哈夫曼编码算法 .....	176
7.9	*并查集和等价关系 .....	177
7.9.1	并查集 ADT .....	177
7.9.2	并查集的存储表示 .....	178
7.9.3	并查集类 .....	178
7.9.4	函数 Union 和 Find .....	179
7.9.5	改进的函数 Union 和 Find .....	179
7.9.6	按等价关系分组 .....	181
	本章小结 .....	181
	习题 .....	182
<b>第 8 章</b>	<b>集合和搜索 .....</b>	<b>184</b>
8.1	集合及其表示 .....	184
8.1.1	集合和搜索的概念 .....	184
8.1.2	动态集 ADT .....	185
8.1.3	集合的表示 .....	186
8.2	顺序搜索 .....	187
8.2.1	无序表的顺序搜索 .....	187
8.2.2	有序表的顺序搜索 .....	188
8.2.3	平均搜索长度 .....	188
8.2.4	自组织表 .....	189

8.3	二分搜索 .....	190
8.3.1	二分搜索算法 .....	190
8.3.2	对半搜索 .....	191
8.3.3	二叉判定树 .....	192
8.3.4	*斐波那契搜索 .....	193
8.4	*搜索算法的时间下界 .....	195
	本章小结 .....	196
	习题 .....	196
<b>第9章</b>	<b>动态集和搜索树 .....</b>	<b>197</b>
9.1	二叉搜索树 .....	197
9.1.1	二叉搜索树的定义 .....	197
9.1.2	二叉搜索树的搜索 .....	198
9.1.3	二叉搜索树的插入 .....	199
9.1.4	二叉搜索树的删除 .....	200
9.1.5	平均情况时间分析 .....	201
9.2	*二叉平衡树 .....	203
9.2.1	二叉平衡树的定义 .....	203
9.2.2	二叉平衡树类 .....	204
9.2.3	二叉平衡树的平衡旋转 .....	205
9.2.4	二叉平衡树的插入 .....	210
9.2.5	二叉平衡树的删除 .....	211
9.2.6	二叉平衡树的高度 .....	214
9.3	*伸展树 .....	215
9.3.1	自调节树和伸展树 .....	215
9.3.2	伸展树的伸展操作 .....	215
9.3.3	伸展树类 .....	217
9.3.4	旋转的实现 .....	218
9.3.5	伸展树的插入运算 .....	218
9.3.6	时间分析 .....	220
	本章小结 .....	222
	习题 .....	222
<b>第10章</b>	<b>多叉搜索树 .....</b>	<b>224</b>
10.1	m叉搜索树 .....	224
10.2	B树 .....	226
10.2.1	B树的定义 .....	226
10.2.2	B树的高度 .....	226
10.2.3	B树的搜索 .....	227
10.2.4	B树的插入 .....	227
10.2.5	B树的删除 .....	229
10.2.6	*B树类 .....	231
10.2.7	*B树的搜索运算 .....	232

10.2.8	*B 树的插入运算 .....	233
10.2.9	*B 树的删除运算 .....	235
10.3	*键树 .....	239
10.3.1	键树的定义 .....	239
10.3.2	双链树 .....	239
10.3.3	Trie 树 .....	240
10.3.4	Trie 树的搜索 .....	242
10.3.5	Trie 树的插入 .....	243
10.3.6	Trie 树的删除 .....	243
10.3.7	Trie 树性能分析 .....	244
	本章小结 .....	244
	习题 .....	244
<b>第 11 章</b>	<b>跳表和散列表</b> .....	<b>246</b>
11.1	字典 .....	246
11.2	*跳表 .....	246
11.2.1	什么是跳表 .....	246
11.2.2	跳表类 .....	248
11.2.3	跳表的搜索 .....	250
11.2.4	跳表的插入 .....	251
11.2.5	跳表的删除 .....	252
11.3	散列表 .....	253
11.3.1	散列技术 .....	253
11.3.2	散列函数 .....	254
11.3.3	拉链法 .....	255
11.3.4	开地址法 .....	256
11.3.5	线性探查法 .....	256
11.3.6	其他开地址法 .....	260
11.3.7	性能分析 .....	262
	本章小结 .....	262
	习题 .....	262
<b>第 12 章</b>	<b>图</b> .....	<b>264</b>
12.1	图的基本概念 .....	264
12.1.1	图的定义与术语 .....	264
12.1.2	图 ADT .....	266
12.2	图的存储结构 .....	267
12.2.1	图的矩阵表示法 .....	267
12.2.2	图的邻接矩阵实现 .....	269
12.2.3	图的邻接表表示法 .....	271
12.2.4	图的邻接表实现 .....	272
12.3	图的遍历 .....	274
12.3.1	扩充的图类 .....	275

12.3.2	深度优先遍历	275
12.3.3	宽度优先遍历	277
12.3.4	基本遍历方法	278
12.4	拓扑排序	279
12.4.1	用顶点代表活动的 AOV 网	279
12.4.2	拓扑排序算法	280
12.4.3	实现拓扑排序算法	281
12.5	*关键路径	283
12.5.1	用边代表活动的 AOE 网	283
12.5.2	关键路径算法	284
12.5.3	实现关键路径算法	286
12.6	最小代价生成树	287
12.6.1	基本概念	287
12.6.2	普里姆算法	287
12.6.3	*克鲁斯卡尔算法	289
12.6.4	*算法正确性	291
12.7	单源最短路径	292
12.7.1	最短路径问题	292
12.7.2	迪杰斯特拉算法	292
12.7.3	数据结构选择	293
12.7.4	实现迪杰斯特拉算法	293
12.8	*所有顶点之间的最短路径	296
12.8.1	弗洛伊德算法	296
12.8.2	实现弗洛伊德算法	297
	本章小结	298
	习题	298
<b>第 13 章</b>	<b>内排序</b>	<b>300</b>
13.1	基本概念	300
13.2	插入排序	301
13.2.1	直接插入排序	301
13.2.2	顺序表直接插入排序	302
13.2.3	*单链表直接插入排序	303
13.2.4	*希尔排序	305
13.3	选择排序	306
13.3.1	简单选择排序	307
13.3.2	*堆排序	308
13.4	交换排序	309
13.4.1	冒泡排序	309
13.4.2	快速排序	311
13.4.3	快速排序算法	311
13.4.4	*快速排序性能分析	313

13.5 两路合并排序 .....	315
13.5.1 合并两个有序序列 .....	315
13.5.2 顺序表两路合并排序 .....	316
13.5.3 *合并排序的递归算法 .....	317
13.5.4 *单链表两路合并排序 .....	317
13.6 *排序算法的时间下界 .....	320
13.7 *基数排序 .....	321
13.7.1 分配排序 .....	321
13.7.2 基数排序算法 .....	322
13.7.3 基数排序 C++程序 .....	323
本章小结 .....	325
习题 .....	325
<b>第 14 章 *文件和外排序 .....</b>	<b>327</b>
14.1 辅助存储器简介 .....	327
14.1.1 主存储器和辅助存储器 .....	327
14.1.2 磁盘存储器 .....	327
14.2 文件 .....	328
14.2.1 文件的基本概念 .....	328
14.2.2 文件的组织方式 .....	329
14.3 文件的索引结构 .....	332
14.3.1 静态索引结构 .....	332
14.3.2 动态索引结构 .....	332
14.4 外排序 .....	333
14.4.1 外排序的基本过程 .....	333
14.4.2 初始游程的生成 .....	334
14.4.3 多路合并 .....	338
14.4.4 最佳合并树 .....	340
本章小结 .....	340
习题 .....	341
<b>附录 A 实习要求和实习题 .....</b>	<b>342</b>
A.1 面向对象方法概述 .....	342
A.2 实习目的 .....	344
A.3 实习要求 .....	344
A.4 实习步骤 .....	344
A.5 实习报告 .....	345
A.6 实习题 .....	346
<b>附录 B 专有名词中英文对照表 .....</b>	<b>349</b>
<b>参考文献 .....</b>	<b>355</b>

# 第1章 基础知识

本章中，首先介绍什么是数据结构，算法和数据结构在计算机学科中的地位如何；然后介绍抽象数据类型概念和面向对象方法，阐明数据结构与两者的关系。本章还简要回顾了 C++ 语言的基本特征；给出了数据结构和算法的描述方法。本章最后讨论算法和算法分析的基本方法。

## 1.1 算法与数据结构

众所周知，计算机由硬件和软件组成，硬件通过软件发挥效用。硬件是躯体，软件是灵魂，而软件的核心是程序。学习程序设计需要掌握一门程序设计语言，它是学习计算机后续课程所必须的技能，但程序设计不等于编码，为了充分利用计算机资源，开发高效的程序，计算机人员还必须掌握计算机学科多方面知识：数据的组织、算法的设计和分析以及软件工程技术等等。

随着计算机科学与技术的发展，计算机应用已远远超出了单纯进行科学计算的范围。从传统的应用领域，如工业控制、情报检索、企业管理、商务处理、图形图像和人工智能等诸多的数据处理领域，发展到电子政务、电子商务、办公自动化、企业资源管理、电子图书馆、远程教育和远程医疗等更广泛的领域。计算机技术已渗透到国民经济的各行各业和人们日常生活的方方面面。

今天，信息技术作为现代技术的标志，已成为世界各国经济增长的主要动力。现实世界中的大量信息都必须转换成数据才能在计算机中存储、处理。数据是信息的载体，应用程序处理各种各样的数据。笼统地说，所谓数据（data），就是计算机加工处理的对象。数据一般分两类：**数值数据（numerical data）**和**非数值数据（non-numerical data）**。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、语音和表格等。这类数据的特点是量大，而且往往有着复杂的内在联系。如果单纯依靠改进程序设计技巧，已无法编制出高效可靠的程序，而必须对数据本身的结构加以研究。数据的组织和表示方法直接影响使用计算机求解问题的效率。算法设计通常建立在所处理的数据的一定组织形式之上。在许多应用中，对于相同数据的同样处理要求，如果选择不同的数据结构，会有不同的处理效率：运算时间和存储空间。数据结构和算法两者是紧密结合的。

对计算机学科来说，数据结构与算法的概念是至关重要的，它们是计算机学科的基础之一，更是软件技术的基础。数据结构与算法之间有着本质的联系。当谈论一种算法时，自然要涉及算法所处理的数据问题；而讨论数据的组织或结构，离开了对处理此类数据的运算及其算法的研究也是无意义的。有人概括过这样一个公式：

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



在计算学科教学计划 1991 (Computing Curricula 1991, CC1991) 中, 数据结构和算法是计算机学科的 9 个研究领域之一。计算学科教学计划 2001 (CC2001) 将其调整成 14 个领域, 数据结构和算法的基本内容主要涵盖在算法与复杂性 (Algorithms and complexity, AL)、程序设计基础 (Programming Fundamentals, PF) 和程序设计语言 (Programming Language, PL) 3 个领域中。CC2001 强调了算法和程序设计, 它建议使用 90 个核心学时讲授这方面的知识, 占整个核心学时数的 32.2%。

数据结构主要是为研究和解决如何使用计算机组织和处理这些非数值问题而产生的理论、技术和方法。

## 1.2 什么是数据结构

### 1.2.1 基本概念

前文提到, 数据是计算机加工处理的对象。数据是由成分数据构成的, 并具有一定的结构。在这里, 称组成数据的成分数据为数据元素 (data element)。通常, 数据元素可以是简单类型的, 如整数、实数、字符等, 也可以是结构类型的, 如记录 (record)。如果把每个学生的记录看作一个数据元素, 它包括学号、姓名、性别等数据项 (data item), 数据项是不可再分割的, 那么, 一个班学生的记录组成了表 1-1 所示的学生情况表。

表 1-1 学生情况表

学号	姓名	性别	其他信息
B02040101	王小红	女	...
B02040102	林悦	女	...
B02040103	陈菁菁	女	...
B02040104	张可可	男	...
⋮	⋮	⋮	⋮

那么, 什么是数据结构? 数据结构是计算机科学与技术领域中广泛使用的术语, 但究竟什么是数据结构, 在计算机科学界至今没有标准的定义。

Sartaj Sahni 在他的《数据结构、算法与应用》一书中称“数据结构是数据对象, 以及存在于该对象的实例和组成实例的数据元素之间的各种联系 (relationship)。这些联系可以通过定义相关的函数来给出”。该书中, Sartaj Sahni 将数据对象 (data object) 定义为“一个数据对象是实例 (instance) 或值 (value) 的集合”。

Clifford A. Shaffer 在《数据结构与算法分析》一书中的定义是: “数据结构是 ADT<sup>①</sup> 的物理实现”。

Lobert L. Kruse 在《数据结构与程序设计》一书中, 将一个数据结构的设计过程分成抽象层、数据结构层和实现层。其中, 抽象层是指抽象数据类型层, 它讨论数据的逻辑结构及其运算, 数据结构层和实现层讨论一个数据结构的表示和在计算机内的存储细节以及运算的实现。

我们认为, 一个数据结构 (data structures) 是由数据元素依据某种逻辑联系组织起来的,

<sup>①</sup> 抽象数据类型 (Abstract Data Type, ADT), 见本章第 1.3 节。