



普通高等教育“十二五”规划教材

数据结构（C语言描述）

主编 任志国



科学出版社

普通高等教育“十二五”规划教材

数据结构(C 语言描述)

主 编 任志国

副主编 蓝才会 赵传成 祁建宏

达文姣 岳秋菊 刘 君

科 学 出 版 社

北 京

内 容 简 介

本书系统地介绍数据结构基础理论知识及算法设计方法,第1~9章从抽象数据类型的角度讨论各种基本类型的数据结构及其应用,主要包括线性表、栈和队列、串、数组和广义表、树和二叉树、图及图算法;第10章和第11章主要讨论查找和排序的各种实现方法及其综合比较;附录给出全书习题中选择、判断、填空题的参考答案。全书采用C语言作为数据结构和算法的描述语言。

本书在内容选取上符合人才培养目标的要求及教学规律和认知规律,在组织编排上体现“先理论、后应用、理论与应用相结合”的原则,并兼顾学科的广度和深度,力求适用面广。本书具有结构严谨、层次清楚、概念准确、深入浅出、描述清晰等特点。

本书可以作为计算机类专业和信息类相关专业的本科教材或考研参考用书,也可以供从事计算机工程与应用工作的科技工作者参考。

图书在版编目(CIP)数据

数据结构: C语言描述 / 任志国主编. —北京: 科学出版社, 2016
普通高等教育“十二五”规划教材
ISBN 978-7-03-049163-3

I. ①数… II. ①任… 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(2016)第 143364 号

责任编辑: 于海云 / 责任校对: 郭瑞芝
责任印制: 徐晓晨 / 封面设计: 迷底书装

科学出版社 出版

北京东黄城根北街16号
邮政编码: 100717
<http://www.sciencep.com>

北京京华虎彩印刷有限公司 印刷

科学出版社发行 各地新华书店经销

*

2016年6月第 一 版 开本: 787×1092 1/16

2016年6月第一次印刷 印张: 21

字数: 550 000

定价: 68.00 元

(如有印装质量问题, 我社负责调换)

前 言

“数据结构”课程是计算机类、电子信息类及相关专业的专业基础课。它在整个课程体系中处于承上启下的核心地位：一方面扩展和深化在离散数学、程序设计语言等课程学到的基本技术和方法；另一方面为进一步学习操作系统、编译原理、数据库等专业知识奠定坚实的理论与实践基础。本课程在教给学生数据结构设计和算法设计的同时，培养学生的抽象思维能力、逻辑推理能力和形式化思维方法，增强分析问题、解决问题和总结问题的能力，更重要的是培养专业兴趣，树立创新意识。本教材在内容选取上符合人才培养目标的要求及教学规律和认知规律，在组织编排上体现“先理论、后应用、理论与应用相结合”的原则，并兼顾学科的广度和深度，力求适用面广。

全书共 11 章。第 1 章综述数据、数据结构和抽象数据类型等基本概念及算法描述与分析方法；第 2~9 章主要从抽象数据类型的角度分别讨论线性表、栈和队列、串、数组和广义表、树和二叉树、图等基本类型的数据结构及其应用；第 10 章和第 11 章讨论查找和排序的各种方法，着重从时间性能、应用场合及使用范围方面进行分析和比较。本书对数据结构众多知识点的来龙去脉做了详细解释和说明；每章后面配有难度各异的习题，并在附录中给出习题的参考答案，供读者理解知识及复习提高之用。全书采用 C 语言描述数据结构和算法。

从课程性质上讲，“数据结构”是高等院校计算机科学、电子信息科学及相关专业教学计划中的一门专业基础课；其教学要求是学会分析研究计算机加工的数据结构的特性，以便为实际应用涉及的数据选择适当的逻辑结构、存储结构及其相应的算法，并初步掌握算法的时空分析技术。从课程学习上讲，“数据结构”的学习是复杂程序设计的训练过程；其教学目的是着眼于原理与应用的结合，在深化理解和灵活掌握教学内容的基础上，学会把知识用于解决实际问题，书写出符合软件工程规范的文件，编写出结构清晰及正确易读的程序代码。可以说，“数据结构”比“高级程序设计语言”等课程有着更高的要求，它更注重培养分析抽象数据的能力。

本书是编者多年从事该课程教学工作的教学成果，编者都是具有副教授以上职称、有 15 年以上该课程教学经验的一线教师。本书由任志国担任主编、赵传成、蓝才会、祁建宏、达文姣、岳秋菊、刘君担任副主编。其中的第 1 章、第 2 章、第 5 章、第 7 章、第 8 章、第 9 章由任志国编写，第 3 章由赵传成编写，第 4 章由岳秋菊编写，第 6 章由达文姣编写，第 10 章由蓝才会编写，第 11 章由祁建宏编写，所有章节习题部分由刘君编写。在本书的构思与编写过程中，得到了安天庆教授、党建武教授、王治和教授的帮助，在算法的实现与调试以及插图的制作过程中，得到了杨业、史淑娟、宗小兵等研究生的帮助，在此表示感谢。

本书可以作为计算机类专业和电子信息类相关专业的教材或考研参考用书，也可供从事计算机工程与应用工作的科技工作者参考。由于作者水平有限，教材中不当之处敬请读者提出批评和建议，编者电子邮件地址：ren_zhiguo@qq.com。

编 者

2016 年 6 月

目 录

前言

第 1 章 绪论	1
1.1 引言	1
1.2 数据结构的基本概念	3
1.2.1 有关概念和术语	4
1.2.2 数据的逻辑结构	4
1.2.3 数据的存储结构	5
1.2.4 数据的运算	5
1.3 数据类型和抽象数据类型	6
1.3.1 数据类型	6
1.3.2 抽象数据类型	7
1.4 算法	8
1.4.1 算法及其特征	8
1.4.2 常见的算法描述方法	9
1.4.3 常见的算法设计方法	10
1.5 算法性能分析与度量	10
1.5.1 时间复杂度	11
1.5.2 空间复杂度	14
1.6 关于学习数据结构	14
1.6.1 数据结构课程的地位	14
1.6.2 数据结构课程体系	15
1.6.3 数据结构课程学习特点	15
习题一	16
第 2 章 线性表	19
2.1 线性表的类型定义	19
2.1.1 线性表的定义	19
2.1.2 线性表的抽象数据类型	20
2.2 线性表的顺序存储及基本操作	21
2.2.1 线性表的顺序存储结构	21
2.2.2 顺序表及相关操作的实现	23
2.2.3 顺序表应用举例	28
2.2.4 线性表顺序存储结构分析	29
2.3 线性表的单链表存储结构	29
2.3.1 线性表的单链表存储结构	30

2.3.2	单链表上相关操作的实现	31
2.3.3	链表应用举例	36
2.3.4	链式存储结构的分析	39
2.4	双链表与其他链式结构	39
2.4.1	线性表的双链表存储结构	39
2.4.2	双链表上相关操作的实现	40
2.4.3	循环链表	43
2.4.4	静态链表	44
2.5	一元多项式的表示及运算	45
2.5.1	一元多项式的表示及存储	45
2.5.2	一元多项式创建与打印	46
2.5.3	一元多项式相加	47
2.5.4	一元多项式相乘	48
习题二		50
第3章	栈	53
3.1	栈的定义及基本运算	53
3.1.1	栈的定义	53
3.1.2	栈的抽象数据类型	53
3.2	顺序栈	54
3.2.1	顺序栈的定义及存储结构	54
3.2.2	顺序栈的基本操作	55
3.3	链栈	57
3.3.1	链栈的定义及存储结构	57
3.3.2	链栈的基本操作	58
3.4	共享栈与多栈	60
3.4.1	共享栈	60
3.4.2	多链栈	63
3.5	栈的应用	65
3.5.1	栈的简单应用	65
3.5.2	栈与递归	70
习题三		71
第4章	队列	74
4.1	队列的定义及基本运算	74
4.1.1	队列的定义	74
4.1.2	队列的抽象数据类型	74
4.2	循环队列	75
4.2.1	循环队列的存储实现	75
4.2.2	循环队列的基本操作	77
4.2.3	动态循环队列	79

4.3	链队列	80
4.3.1	链队列的定义及存储结构	80
4.3.2	链队列的基本操作	81
4.4	队列的其他存储结构	82
4.4.1	循环多队列	82
4.4.2	动态循环多队列与链式多队列	84
4.5	队列的应用	84
	习题四	85
第5章	串	88
5.1	串的定义及其基本运算	88
5.1.1	串的定义	88
5.1.2	串的抽象数据类型	88
5.2	串的定长顺序存储	90
5.2.1	定长顺序存储的定义	90
5.2.2	定长顺序串的基本运算	90
5.3	串的模式匹配算法	95
5.3.1	简单模式匹配算法——BF 算法	95
5.3.2	改进的模式匹配算法——KMP 算法	96
5.4	串的堆存储结构	100
5.4.1	堆存储结构的定义	100
5.4.2	基于堆结构的基本运算	101
5.5	串的块链存储结构	104
5.5.1	块链存储结构的定义及其存储结构	104
5.5.2	基于块链结构的基本运算	105
5.6	串的应用	113
	习题五	113
第6章	数组和广义表	116
6.1	数组的概念和存储	116
6.1.1	数组的概念	116
6.1.2	数组的存储结构	117
6.2	特殊矩阵的压缩存储	121
6.2.1	对称矩阵的压缩存储	121
6.2.2	三角矩阵的压缩存储	122
6.2.3	带状矩阵的压缩存储	123
6.3	稀疏矩阵的压缩存储	124
6.3.1	稀疏矩阵的三元组表存储	125
6.3.2	稀疏矩阵的十字链表存储	128
6.4	广义表	133
6.4.1	广义表的基本概念	133

6.4.2	广义表的基本运算	135
6.4.3	广义表的存储结构	136
6.4.4	广义表上的基本算法	138
习题六		140
第7章 二叉树和树		144
7.1	二叉树的定义与性质	144
7.1.1	二叉树的基本概念	144
7.1.2	二叉树的主要性质	146
7.1.3	二叉树的抽象数据类型	147
7.2	二叉树的存储结构及创建	148
7.2.1	顺序存储结构	148
7.2.2	二叉树的链式存储结构	150
7.2.3	二叉树的创建算法	151
7.3	二叉树的遍历及应用	152
7.3.1	二叉树的遍历	152
7.3.2	二叉树遍历的非递归实现	154
7.3.3	遍历算法的应用	158
7.3.4	由遍历序列恢复二叉树	159
7.4	线索二叉树	161
7.4.1	线索二叉树的定义及结构	161
7.4.2	线索二叉树的创建及遍历	163
7.4.3	线索二叉树的其他相关算法	165
7.5	哈夫曼树及其应用	169
7.5.1	哈夫曼树的基本概念	169
7.5.2	构造哈夫曼树	171
7.5.3	哈夫曼编码	172
7.5.4	哈夫曼树的应用	174
7.6	树的概念与表示	175
7.6.1	树的定义及相关术语	175
7.6.2	树的表示	176
7.6.3	树的存储	177
7.7	树与二叉树的转换	180
7.7.1	树或树林转换为二叉树	180
7.7.2	二叉树转换为树或树林	181
7.7.3	树或树林的遍历	183
7.8	树的应用	183
7.8.1	判定树	183
7.8.2	集合的表示	185
习题七		186

第 8 章	图论	192
8.1	图的基本概念	192
8.1.1	图的定义	192
8.1.2	图的相关术语	193
8.1.3	图的抽象数据类型	193
8.2	图的邻接表存储结构	195
8.2.1	图的邻接表存储结构定义	195
8.2.2	建立在图的邻接表存储结构的基本算法	196
8.2.3	创建图的邻接表存储结构	203
8.3	图的邻接矩阵存储结构	205
8.3.1	图的邻接矩阵存储结构定义	205
8.3.2	建立在图的邻接矩阵存储结构的基本操作算法	207
8.3.3	创建图的邻接矩阵存储结构	212
8.4	图的其他存储结构	214
8.4.1	图的十字链表存储结构	214
8.4.2	图的邻接多重表存储结构	215
8.5	图的广度优先遍历	216
8.5.1	广度优先搜索	216
8.5.2	邻接矩阵存储结构上的 BFS 算法	217
8.5.3	邻接表存储结构上的 BFS 算法	218
8.6	图的深度优先遍历	219
8.6.1	深度优先搜索	219
8.6.2	邻接矩阵存储结构上的 DFS 算法	220
8.6.3	邻接表存储结构上的 DFS 算法	221
	习题八	223
第 9 章	图算法及应用	226
9.1	最小生成树	226
9.1.1	最小生成树的定义	226
9.1.2	构成最小生成树的 Prim 算法	227
9.1.3	构成最小生成树的 Kruskal 算法	230
9.2	最短路径	230
9.2.1	求图中某一顶点到其余各顶点的最短路径——Dijkstra 算法	231
9.2.2	每一对顶点之间的最短路径——Floyd 算法	233
9.3	AOV 网的应用	236
9.3.1	AOV 网的定义	236
9.3.2	拓扑排序	238
9.4	AOE 网的应用	240
9.4.1	AOE 网的定义	240
9.4.2	关键路径	241
	习题九	247

第 10 章 查找	252
10.1 查找的基本概念	252
10.2 静态查找表	253
10.2.1 顺序查找	253
10.2.2 有序表的折半查找	254
10.2.3 分块查找	256
10.3 二叉排序树	256
10.3.1 二叉排序树的定义	256
10.3.2 二叉排序树的相关算法	257
10.3.3 二叉排序树的查找效率分析	260
10.4 平衡二叉排序树	260
10.4.1 平衡二叉排序树的定义	260
10.4.2 调整不平衡的二叉排序树	261
10.4.3 创建平衡二叉排序树	263
10.5 其他查找树	264
10.5.1 B 树及其基本操作	264
10.5.2 B+树的基本概念	267
10.6 散列表	268
10.6.1 散列表的基本概念	268
10.6.2 散列函数的设计	268
10.6.3 冲突的处理方法	270
10.6.4 散列表的查找分析	272
习题十	273
第 11 章 排序	280
11.1 排序的基本概念	280
11.1.1 排序的定义	280
11.1.2 排序方法的分类	281
11.1.3 排序算法的分析方法	281
11.2 插入排序	281
11.2.1 直接插入排序	282
11.2.2 折半插入排序	283
11.2.3 希尔排序	285
11.3 交换排序	287
11.3.1 冒泡排序	287
11.3.2 快速排序	289
11.4 选择排序	291
11.4.1 简单选择排序	291
11.4.2 树形选择排序	293
11.4.3 堆排序	294

11.5 归并排序.....	297
11.6 基数排序.....	299
11.6.1 多关键码排序.....	299
11.6.2 链式基数排序.....	300
11.7 内部排序算法的比较	304
11.7.1 内部排序算法的比较	304
11.7.2 内部排序算法的选用	305
习题十一	306
附录一 习题参考答案	312
附录二 学期考试样卷	318
参考文献	323

第1章 绪 论

用计算机求解任何问题都离不开程序设计，而程序设计的实质是数据表示和数据处理。数据要能被计算机处理，首先要能够被存储在计算机的内存中，这项任务称为数据表示，数据表示的核心任务是数据结构的设计。一个实际问题的求解必须满足各项处理要求，这项任务称为数据处理，数据处理的核心任务是算法设计。数据结构课程主要讨论数据表示和数据处理的基本问题。本章概括地介绍数据结构的基本概念、基本思想和基本方法。

学习要点：

- 数据结构的研究内容、数据结构相关的概念和术语。
- 数据结构的三要素：逻辑结构、物理结构和数据运算。
- 算法及算法的时间复杂度和空间复杂度的分析与计算。

1.1 引 言

自1946年世界上第一台计算机诞生以来，计算机产业飞速发展的速度远远超出了人们对它的预料。现在计算机的应用已经渗透到人类生活的各个领域。如今，计算机的应用不再局限于科学计算，而更多地用于过程控制、管理、信息处理等非数值计算领域。计算机加工处理的对象也由数值发展到字符、表格、图形、图像、音频、视频、动画等具有一定结构的数据，这就给程序设计带来一个新的课题。为了编写出一个好的程序，必须分析待处理对象的特性及各处理对象之间存在的关系。这就是“数据结构”这门学科形成和发展的背景。

目前，计算机更多的是用于过程控制、事务管理、信息处理等非数值计算领域。计算机要进行信息处理首先要进行信息的表示，信息的表示和组织又直接关系到处理信息的程序的效率。随着应用问题的不断复杂，导致信息量剧增与信息范围拓宽，使许多系统程序和应用程序的规模很大，结构又相当复杂。因此，必须分析待处理问题中对象的特征及各对象之间存在的关系，这就是“数据结构”这门学科所要研究的问题。

一般来说，用计算机解决一个具体问题时，一般需要经过如下几个步骤：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求解此数学模型的数据结构和算法，接着编写程序进行调试、测试，最后运行该程序，直至得到最终结果。

很多问题的求解最后都转化为求解数学方程或数学方程组，即使是不需要计算机求解的简单问题也需要一个数学模型来描述。例如，大家熟悉的“鸡兔同笼”问题可转化为二元一次方程组进行求解。又如，在房屋设计或桥梁设计中的结构应力分析计算问题可化解为线性代数方程组求解的问题。再如，对于天天看到的天气预报，它的数学模型是一个环流模式方程。如果读者对此类的数学问题感兴趣，可以学习《计算方法》等课程。这些问题不作为数据结构所讨论的问题。

数据结构主要讨论非数值计算问题，如人们在日常生活中遇到的下面一些问题。

(1) 文档的编辑处理、对一个班级学生信息的管理等问题。用编辑器处理文档，其实质是

对字符串的处理问题。班级学生信息的管理问题主要考虑如何根据基本的操作有效地组织数据、存储数据。

(2) 八皇后问题。在八皇后问题中，处理过程不是根据某种确定的计算法则，而是利用试探和回溯的探索技术求解。为了求得合理布局，在计算机中要存储布局的当前状态。从最初的布局状态开始，一步步地进行试探，每试探一步形成一个新的状态，整个试探过程形成了一棵隐含的状态树。如图 1.1 所示(为了描述方便，将八皇后问题简化为四皇后问题)。

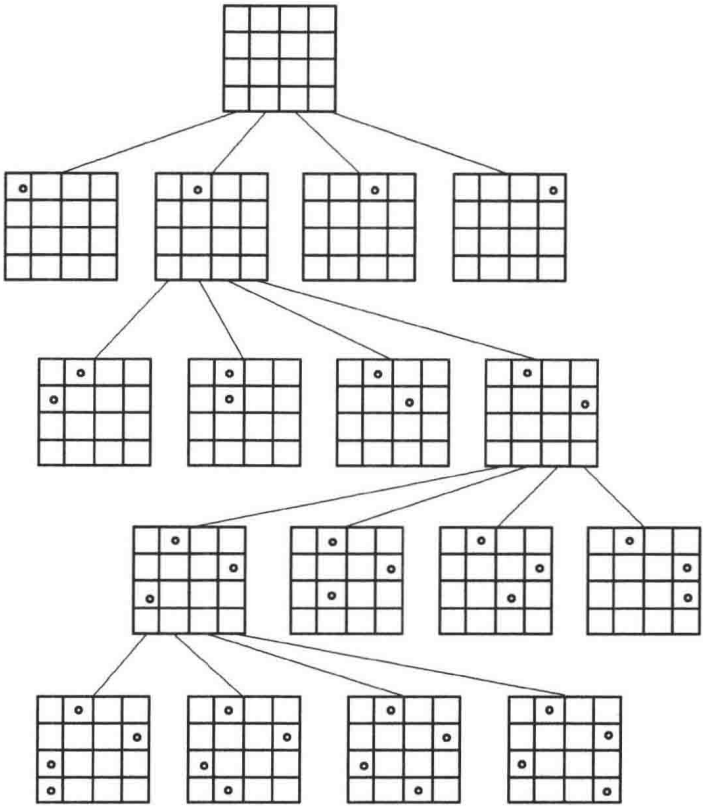


图 1.1 四皇后问题中隐含的状态树

回溯法求解过程实质上就是一个遍历状态树的过程。在这个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。

(3) 多叉路口信号灯的设置问题。通常，在多叉路口只需设红绿两色的交通灯便可以保证正常的交通秩序，而在多叉路口须设计几种颜色的交通灯才能既使车辆相互之间不碰撞，又能达到车辆的最大流量呢？

如图 1.2 所示的五叉路口，其中 C 和 E 为单行道，在路口有 13 条可行的通路(A→B、A→C、A→D、B→A、D→C、E→D、B→C、B→D、E→A、D→A、D→B、E→B、E→C)，其中有的可以同时通行，如 A→B 和 E→C，而有的不能同时通行，如 E→B 和 A→D。那么，在路口应如何设置交通灯进行车辆的管理呢？

通常，这类交通、道路问题的数学模型是一种称为“图”的数据结构。如图 1.3 所示，图中每个圆圈(又称为顶点)表示五叉路口上的一条通路，两个圆圈之间的连线(又称为边)表示两个圆圈所代表的通路，不能同时通行，则设置交通灯的问题等价于对图的顶点的染色问题，要求对图上的每个顶点染一种颜色，并且要求有线相连的两个顶点不能具有相同的颜色，

而总的颜色种类应尽可能地少。如图 1.3 所示为一种染色结果，圆圈中的数字表示交通灯的不同颜色。

- 1 号色灯：A→B、A→C、A→D、B→A、D→C、E→D。
- 2 号色灯：B→C、B→D、E→A。
- 3 号色灯：D→A、D→B。
- 4 号色灯：E→B、E→C。

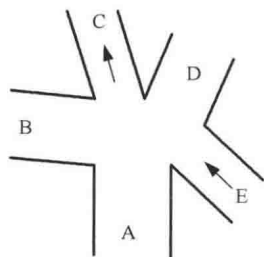


图 1.2 多叉路口

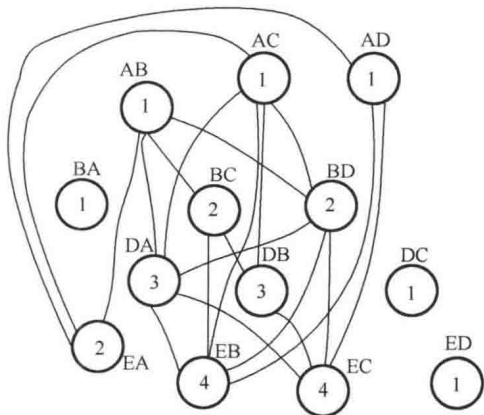


图 1.3 多叉路口各方向通路

综上所述，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，这些问题就是数据结构所研究的内容。

所以，可以直观地认为数据结构是一门研究非数值计算的程序设计问题中计算机的操作对象，以及它们之间的关系、操作、存储等问题的学科。

1968 年，美国唐纳德·克努特 (Donald Ervin Knuth) 教授开创数据结构的最初体系。他的数百万字的多卷本《计算机程序设计的艺术》(The Art of Computer Programming) 堪称计算机科学理论与技术的经典巨著。1968 年出版的该巨著第一卷《基本算法》是第一本较系统地阐述数据的逻辑结构、存储结构及其操作的著作。该巨著的第一卷和 1973 年出版的第三卷《排序与搜索》中的一些基本内容就构成了数据结构的最初体系。由于唐纳德·克努特对计算机科学的卓越贡献，他本人获得了 1974 年的图灵奖。

数据结构从诞生到现在，在不到半个世纪的时间里形成了坚实的理论基础和广泛的应用领域，吸引越来越多的研究者加入。数据结构的诞生和发展给计算机信息管理带来了一场巨大的革命。同时，随着应用的扩展与深入，数据结构的研究领域也已经大大地拓广和深化。

1.2 数据结构的基本概念

数据结构是计算机科学与技术专业最重要的专业基础核心课程。所有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实际问题，仅掌握几种计算机程序设计语言是难以应付众多复杂问题的。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。打好数据结构这门课程的扎实基础，对于学习计算机专业的其他课程，如算法设计与分析、操作系统、编译原理、数据库管理系统、软件工程、人工智能等都是非常重要的。

1.2.1 有关概念和术语

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定义。

数据(Data):是信息的载体,是描述客观事物属性的数字、字符及所有能够输入到计算机中并被计算机识别、存储和加工处理的符号的集合。

数据是计算机程序加工的“原料”,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。

数据项(Data Item):是数据中不可分割,且具有独立含义的最小单位,数据元素是数据项的集合。

数据元素(Data Element):是数据的基本单位,通常作为一个整体进行考虑和处理。一个数据元素可以由若干个数据项来组成,数据项是构成数据元素不可分割的最小单位。在有些情况下,数据元素也称为元素、结点、顶点和记录。例如,学生记录就是一个数据元素,由学号、姓名、性别等数据项组成。

数据对象(Data Object):是具有相同性质的数据元素的集合,是数据的一个子集。

数据结构(Data Structure):是指互相之间存在一种或多种关系的数据元素的集合。数据结构包括逻辑结构、存储结构和数据的运算三方面的内容(也称为数据结构三要素)。数据的逻辑结构和存储结构是密不可分的两个方面,一个算法的设计取决于所选定的逻辑结构,而算法的实现依赖所采用的存储结构。

1.2.2 数据的逻辑结构

在任何问题中,数据元素之间都不会是孤立的,它们之间都存在这样或那样的关系,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,通常有集合结构、线性结构、树形结构、图形结构等4种基本的逻辑结构,如图1.4所示。

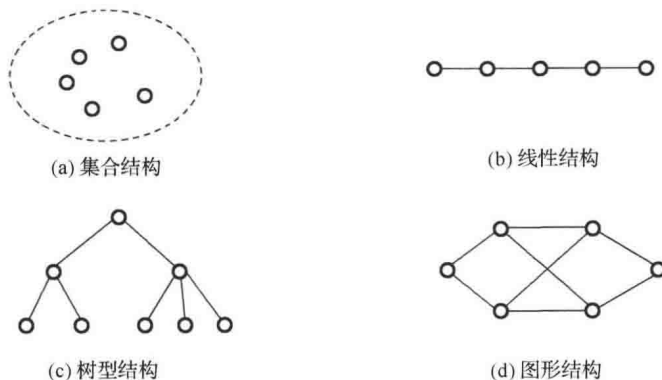


图 1.4 数据的逻辑结构

(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。

(2) 线性结构: 该结构的数据元素之间存在一对一的关系。

(3) 树型结构: 该结构的数据元素之间存在一对多的关系。

(4) 图形结构: 该结构的数据元素之间存在多对多的关系,图形结构也称为网状结构。

数据的逻辑结构又可分为线性结构和非线性结构。线性结构又包括一般线性表、操作受限线性表(栈、队列、串)和线性表推广(数组、广义表)。非线性结构又包括集合、树形结构和图形结构。

1.2.3 数据的存储结构

数据结构在计算机中的表示(又称为映像)称为数据的物理结构,又称为存储结构。它不同于逻辑结构,是依赖计算机语言的,是具体的。通常,一个数据元素在计算机内用一块连续的存储单元来表示。那么,在计算机中怎样存储表中所有的数据元素呢?数据结构一般用下面四种基本的存储结构来表示数据元素之间的关系。

1. 顺序存储结构

该方法把逻辑上相邻的数据元素存储在物理位置也相邻的存储单元里,数据元素之间的逻辑关系由存储单元的邻接关系来体现,由此得到的存储表示称为顺序存储结构。顺序存储结构主要应用于线性结构,非线性结构也可以通过某种线性的方法实现顺序存储。其优点是可实现随机存取,每个元素占用最少的存储空间,即存储密度大。其缺点是只能使用相邻的一整块存储单元,因此可能产生较多的外部碎片。

2. 链式存储结构

该方法不要求逻辑上相邻的数据元素在物理位置上也相邻,数据元素之间的逻辑关系由附加的指针表示,由此得到的存储表示称为链式存储结构。其优点是不会出现碎片现象,可充分利用所有存储单元。其缺点是每个元素因存储指针而占用额外的存储空间,并且只能实现顺序存取。

3. 索引存储结构

该方法通常在存储数据元素信息的同时,还建立附加的索引表。索引表由若干索引项组成。索引项的一般形式是:(关键字、地址)。关键字(Key)是能唯一标识一个数据元素的那些数据项。其优点是检索速度快。缺点是增加附加的索引表占用较多的存储空间。在增加和删除数据时要修改索引表,因而花费较多的时间。

4. 哈希(散列)存储结构

该方法的基本思想是根据数据元素的关键字直接计算出该数据元素的存储地址。其优点是检索、增加、删除结点的操作都很快。缺点是如果散列函数不好,可能出现数据元素存储地址的冲突,而解决冲突会增加时间和空间开销。

这四种基本存储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映像。同一逻辑结构采用不同的存储方法,可以得到不同的存储结构;采用不同的存储结构,其数据处理效率往往不同。选择何种存储结构来表示相应的逻辑结构,视具体要求而定,主要考虑运算方便及算法的时空要求。

1.2.4 数据的运算

为了有效地处理数据,可将数据按一定的逻辑结构组织起来,并选择适当的存储方法存储数据,然后再对数据进行运算。

数据的运算是定义在数据的逻辑结构之上的，每一种逻辑结构都有一个运算的集合，并指出运算的功能，例如，查找、插入、删除、修改等，这些运算实际上是在数据元素上施加的一系列的抽象操作。所谓抽象操作，是只知道这些操作要求“做什么”，而无须考虑“如何做”，只有在确定了存储结构之后，才考虑如何具体实现这些运算。下面介绍几种常见的数据运算。

- (1) 建立(Create)一个数据结构；
- (2) 消除(Cancel)一个数据结构；
- (3) 从一个数据结构中删除>Delete)一个数据元素；
- (4) 把一个数据元素插入(Insert)到一个数据结构中；
- (5) 对一个数据结构进行访问(Visit)；
- (6) 对一个数据结构中的数据元素进行修改(Modify)；
- (7) 对一个数据结构进行排序(Sort)；
- (8) 对一个数据结构进行查找(Search)。

由于数据的运算是建立在数据的逻辑结构之上的，因此，某一种具体的数据结构，除了这些基本运算之外，根据实际问题还有其特有的一些运算。例如：在城市交通问题中，求两城市之间最短路线的算法；判断从城市的任一点出发乘坐公共汽车是否可以达到城市任何地方的运算；在家族中如何找到某个人的兄弟等运算；在电话号码管理问题中如何高效快速查找相关联系人等运算。所以，数据的其他运算是除了基本运算之外根据处理的问题“灵活”出现的。在数据结构的教学中一般只讨论那些基本运算。

数据结构的发展趋势包括两个方面：一是面向专门领域中特殊问题的数据结构的研究和发展，如图形数据结构、知识数据结构、空间数据结构；二是从抽象数据类型角度出发，用面向对象的观点来讨论数据结构。

1.3 数据类型和抽象数据类型

1.3.1 数据类型

“数据类型”是和数据结构密切相关的一个概念。它最早出现在高级程序设计语言中，用以刻画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。数据类型显性或隐性规定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(Data Type)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。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结构的，也可以是非结构的。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。