



- 遵循“循序渐进”和“理论联系实际”的教学原则
- 注重内容的实用性，培养学生的专业能力
- 配套电子教案、程序源代码、习题答案等资源
- 适合高等院校应用型本科及高职高专计算机类各专业

数据结构

(C语言版)

肖宏启 主 编
刘昌明 廖银花 副主编
魏怀明 主 审

配备
素材、电子教案



21 世纪计算机系列规划教材

数 据 结 构

(C 语言版)

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书对常用的数据结构做了系统的介绍,既注重原理又强调实践,配有大量的图表和习题,概念讲解清晰、逻辑性强、可读性好。主要内容包括:数据结构的基本概念;算法描述和算法分析初步;线性表、栈、队列、串和广义表、树、图等结构;排序和查找的各种方法。每章均例举了典型应用实例,并配有算法和相应源程序以供教学和实践使用。另外还在附录中设计了上机实验题供读者独立完成,以加深读者对所学知识的理解和应用。

作为“21 世纪计算机系列规划教材”之一,本书可作为高等院校应用型本科及高职高专院校计算机专业教材,也可以作为大学非计算机专业学生的选修课教材和广大从事计算机应用的科技人员的参考书。

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

数据结构: C 语言版/肖宏启主编. —北京:电子工业出版社, 2009.12

(21 世纪计算机系列规划教材)

ISBN 978-7-121-09918-2

I. 数… II. 肖… 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.TP311.12 TP312

中国版本图书馆 CIP 数据核字(2009)第 211735 号

责任编辑:程超群

印 刷:北京季蜂印刷有限公司

装 订:三河市万和装订厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:19.5 字数:492.8 千字

印 次:2010 年 1 月第 1 次印刷

印 数:3 000 册 定价:30.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前 言

“数据结构”是计算机程序设计的重要理论基础，是计算机及其应用专业的一门重要基础课程和核心课程。它不仅是学习后继软件专业课程的先导，而且已成为其他工科类专业的热门选修课程。本教材列入“21 世纪计算机系列规划教材”，主要面向应用型本科及高职高专院校计算机类专业的学生，培养技术应用型人才。教材内容的构造力求体现“以应用为主体”，强调对理论知识的理解和运用，实现高校应用型本科及高职高专教学以实践体系为主和以技术应用能力培养为主的目标。

本书共分 9 章。第 1 章阐述数据、数据结构和算法等基本概念；第 2~7 章分别讨论了线性表、栈、队列、串和广义表、树和二叉树以及图等基本数据结构及其应用；第 8~9 章讨论查找和排序的各种实现方法及实用分析。

长期以来，由于数据结构课程自身的抽象性和严密性，教师大都感觉数据结构课程难教，学生普遍反映数据结构课程难学，学生很难独立完成算法的实现。基于上述问题，我们在编写本教材时充分考虑了学生的知识结构和教师的教学方法。本教材的编写宗旨是：既注重原理又注重实践，既注重抽象思维又注重形象思维，既方便自学又方便教学。每章均有大量应用举例及分析和一个实验子系统及一定量的习题，并在附录里增加了上机实验题供学生独立完成。

本教材有以下特点：

(1) 对基础理论知识的阐述由浅入深、通俗易懂，内容组织和编排以应用为主线。

(2) 各章均列举分析了很多实用的例子，这有助于学生加深对基础理论知识的理解和实际应用能力的培养。

(3) 考虑到此课程的先导课程是“C 语言程序设计”，书中算法均采用可在计算机上运行的 C 语言程序来描述。这样，降低了算法设计的难度，使学生能更直观形象地理解这些算法。书中提供的实现代码均在 VC 6.0 中编译通过，并给出了所有程序的运行结果。若教师教学采用 TC 环境，只需在相应实现代码中加上相关头文件即可。

(4) 为配合本教材的教学，还编制了多媒体课件，对加深理解基本概念起到更直观的效果。多媒体课件、书中所有算法及实验子系统源代码和习题答案可从电子工业出版社的华信教育资源网 (<http://www.hxedu.com.cn>) 下载，或通过 E-mail 向肖宏启老师索取：xiaohongqi2000@163.com。

(5) 在教材中使用“▲思考”标志，提出问题拓展学生思维，在教学中恰到好处地启发学生的思维。

(6) 为避免 C 语言中数组的第一个元素的下标为 0 给学习和讲授带来的不便，本书在没有特别声明的地方均不使用 C 语言中数组下标为 0 的元素。

本书由肖宏启整体构思，在多位教师长期从事数据结构课程教学的经验基础上，经多次反复磋商和共同讨论定稿，是多位作者共同合作的产物。魏怀明副教授编写了第 5 章并详细审阅了全书，韦军博士审阅了该书并提供了许多宝贵意见，陈元春副教授提供了本书的大部分习题，本书编写过程中参考了许多作者的大量文献资料和国内外优秀教材，电子工业出版社对本书的出版给予了大力支持和帮助，作者谨此一并致以诚挚的谢意。

本书的第4章由冯韵编写，第6章由廖银花编写，第7章及附录部分由刘昌明编写，其余章节均由肖宏启编写。全书由肖宏启统稿、修改。

本教材讲课时数可为60~72学时。上机时数可灵活安排。教师可根据学时数、专业和学生的实际情况选讲应用举例中一些较难的例子。

由于编写教材的时间紧张，难免存在疏漏，敬请读者批评指正。

编 者
2009年9月

目 录

第 1 章 绪论	1
1.1 什么是数据结构	2
1.1.1 数据结构研究的内容	2
1.1.2 学习数据结构的必要性	3
1.2 数据的逻辑结构	4
1.2.1 基本概念	4
1.2.2 逻辑结构的描述	4
1.3 数据的存储结构	6
1.4 算法和算法分析	7
1.4.1 算法特性	7
1.4.2 影响算法效率的因素	8
1.4.3 算法效率的评价	8
实验 1	11
习题 1	11
第 2 章 线性表	15
2.1 线性表的定义与运算	16
2.1.1 线性表的定义	16
2.1.2 线性表的基本操作	17
2.2 线性表的顺序存储	18
2.2.1 顺序表	18
2.2.2 顺序表上基本运算的实现	19
2.3 线性表的链式存储	24
2.3.1 线性链表	24
2.3.2 线性链表上基本运算的实现	26
2.3.3 循环链表	38
2.3.4 双向链表	39
2.3.5 顺序表和链表的比较	41
2.4 线性表的应用举例	42
实验 2 线性表子系统	49
习题 2	50
第 3 章 栈	53
3.1 栈的定义和运算	54
3.1.1 栈 (Stack) 的定义	54
3.1.2 栈的运算	54
3.2 栈的存储和实现	55

3.2.1 顺序栈	55
3.2.2 链栈	61
3.3 栈的应用举例	65
实验3 栈子系统	72
习题3	73
第4章 队列	75
4.1 队列的定义和基本运算	76
4.1.1 队列 (Queue) 的定义	76
4.1.2 队列的基本运算	76
4.2 队列的存储实现及运算实现	77
4.2.1 顺序队列	77
4.2.2 链队列	85
4.3 队列应用举例	91
实验4 队列子系统	92
习题4	92
第5章 串和广义表	95
5.1 串的定义和基本运算	96
5.1.1 串的定义	96
5.1.2 串的基本运算	96
5.2 串的实现	97
5.2.1 定长顺序存储	97
5.2.2 链式存储	98
5.2.3 堆分配存储	99
5.3 串基本运算的实现	101
5.4 广义表	114
5.4.1 广义表的定义和性质	114
5.4.2 广义表的存储	115
5.5 串的应用举例	117
实验5 串子系统	119
习题5	119
第6章 树和二叉树	123
6.1 树的定义和术语	124
6.1.1 树的定义	124
6.1.2 基本术语	125
6.1.3 树的存储	126
6.2 二叉树	129
6.2.1 二叉树的定义	129
6.2.2 二叉树的性质	130
6.2.3 二叉树的存储	132

6.2.4	二叉树的建立	134
6.3	遍历二叉树	136
6.3.1	遍历二叉树	136
6.3.2	恢复二叉树	141
6.4	二叉树的转换	143
6.4.1	一般树转换为二叉树	143
6.4.2	森林转换为二叉树	145
6.4.3	二叉树转换为树和森林	145
6.5	二叉树的应用举例	146
6.6	哈夫曼树及其应用	149
6.6.1	哈夫曼树的引入	149
6.6.2	哈夫曼树的建立	151
6.6.3	哈夫曼编码	152
实验 6	树子系统	157
习题 6		157
第 7 章	图	163
7.1	图的定义和术语	164
7.1.1	图的定义	164
7.1.2	图的相关术语	164
7.1.3	图的基本操作	168
7.2	图的存储表示	168
7.2.1	邻接矩阵	168
7.2.2	邻接表	171
7.2.3	边集数组	174
7.3	图的遍历	175
7.3.1	深度优先搜索	176
7.3.2	广度优先搜索	181
7.4	生成树和最小生成树	188
7.4.1	生成树和生成森林	188
7.4.2	最小生成树	189
7.4.3	普里姆 (Prim) 算法	190
7.4.4	克鲁斯卡尔 (Kruskal) 算法	194
7.5	单源最短路径	198
7.5.1	单源最短路径的概念	198
7.5.2	求单源最短路径的方法	198
7.6	AOV 网与拓扑排序	203
7.6.1	AOV 网与拓扑排序	203
7.6.2	拓扑排序的实现	204
7.7	AOE 网与关键路径	208

7.7.1	AOE 网与关键路径	208
7.7.2	关键路径的确定	209
实验 7	图子系统	209
习题 7		210
第 8 章	查找	213
8.1	查找的基本概念	214
8.2	静态查找表	215
8.2.1	顺序查找	216
8.2.2	二分查找	218
8.2.3	分块查找	223
8.3	动态查找表	224
8.3.1	二叉排序树定义	224
8.3.2	二叉排序树的插入和生成	225
8.3.3	二叉排序树的删除	227
8.3.4	二叉排序树上的查找	230
8.4	散列表	236
8.4.1	散列表与散列函数	236
8.4.2	散列函数的构造方法	237
8.4.3	处理冲突的方法	239
8.4.4	散列表的查找及分析	242
实验 8	查找子系统	246
习题 8		246
第 9 章	排序	251
9.1	概述	252
9.2	插入排序	253
9.2.1	直接插入排序	253
9.2.2	二分插入排序	256
9.2.3	希尔排序	258
9.3	交换排序	261
9.3.1	冒泡排序	261
9.3.2	快速排序	266
9.4	选择排序	270
9.4.1	简单选择排序	270
9.4.2	堆排序	273
9.5	归并排序	278
9.6	各种排序方法的比较	282
实验 9	排序子系统	282
习题 9		283

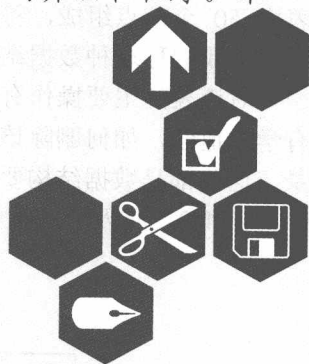
附录 上机实验题	287
实验 1 线性表	288
实验 2 栈	289
实验 3 队列	290
实验 4 树及二叉树	292
实验 5 图	293
实验 6 查找	296
实验 7 排序	297
参考文献	300

1

绪 论

本章内容概要:

计算机科学是一门研究数据表示和数据处理的科学。计算机在发展初期,其应用范围是数值计算,所处理的数据都是整型、实型、布尔型等简单数据,并以此为加工对象进行数值型程序的设计。后来,随着电子技术的发展,计算机逐渐进入到商业、制造业等其他领域,从而广泛地应用于数据过程和过程控制。与此相应,计算机处理的数据也不再是简单的数值,而是字符串、图像、图形、声音、视频等复杂的数据。数据是计算机化的信息,它是计算机可以直接处理的最基本和最重要的对象。无论是在进行科学计算或数据处理、过程控制,还是在对文件的存储和检索及数据库技术等计算机应用领域中,都是对数据进行加工处理的过程。因此,要设计出一个结构好、效率高的程序,必须研究数据的特性和数据间的相互关系及其对应的存储表示,数据结构就是一门研究这些问题的课程,并利用这些特性和关系设计出相应的算法和程序。本章将介绍数据结构和算法分析的基本概念。



1.1 什么是数据结构

1.1.1 数据结构研究的内容

在计算机发展的初期,人们使用计算机的目的主要是处理数值计算问题。当我们使用计算机来解决一个具体问题时,一般需要经过下列几个步骤:首先要从该具体问题抽象出一个适当的数学模型,然后设计或选择一个解此数学模型的算法,最后编制程序、运行并调试程序,直到实际问题被解决。

由于当时所涉及的运算对象是简单的整型、实型或布尔类型数据,所以程序设计者的主要精力是集中于程序设计的技巧上,而无须重视数据结构。随着计算机软、硬件的发展和应用领域的不断扩大,计算机处理的对象更多地是非数值计算问题,这类问题涉及的数据结构更为复杂,数据元素之间的相互关系一般无法用数学方程式加以描述,此时解决这类问题的关键不再是数学分析和计算方法,而是必须建立相应的数据结构来进行描述,分析问题中所用到的数据是如何组织的,研究数据之间的关系如何,进而设计出合适的数据结构,才能有效地解决问题。下面所列举的就是属于这一类的具体问题,用以说明什么是数据结构和数据结构所研究的内容。

例1.1 某班级学生学籍档案的管理,如表1.1所示。

表 1.1 某班级学生学籍档案信息

学号	姓名	性别	年龄	入学成绩
2009001	张 三	男	21	80
2009002	刘薇薇	女	23	90
2009003	马 琳	男	22	78
⋮	⋮	⋮	⋮	⋮
2009050	李 明	男	22	85

在这个班级学生学籍档案信息表中,共有 50 个学生。我们可以把表中每个学生的信息看成一条记录并称之为一个节点,表中的每个节点由 5 个数据项组成。该学生学籍档案信息表由 50 个节点组成,每个节点排列有先后次序,形成一种线性关系。这是一种典型的数据结构,我们称这种数据结构为线性表。

对该表的主要操作有:在给出学号时,如何在表中快速查找到所对应的学生的信息;若有学生退学,如何删除该学生的记录;若有新生入学该班级时,如何在表中插入一条新记录。这些都是数据结构要研究的内容。

例 1.2 航天学院教学行政机构示意图,如图 1.1 所示。

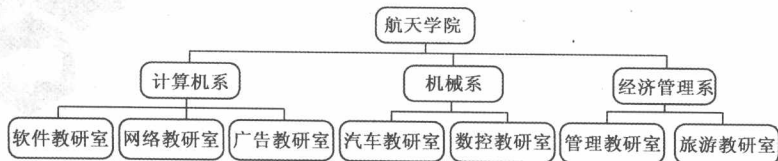


图 1.1 航天学院教学行政机构示意图

对于航天学院的教学行政机构,可以把该学院的名称看成树根,把下设的若干个系看成它的树枝中间节点,把每个系的教研室看成树叶。树中的每个节点可以包含较多的信息,节点之间的关系不再是顺序的,而是分层、分叉的一对多的非线性结构。这也是我们常用的一种数据结构,我们称之为树形结构,如图 1.1 所示。树形结构的主要操作有遍历、查找、插入和删除等。

例 1.3 城市之间建立通信网络的问题,如图 1.2 所示。

在 n 个城市之间建立通信网络,要求在任意两个城市之间都有直接或间接的通信线路;在已知某些城市之间直接通信线路预算造价的情况下,使网络造价最低。

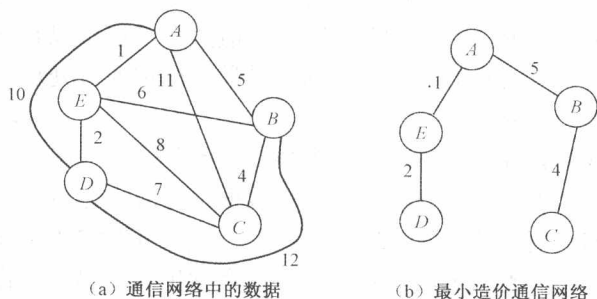


图 1.2 用图描述通信网络问题

当 n 很大时,这样的问题只能用计算机来求解。我们用图 1.2 (a) 中描述的关系来说明:图中的每个小圆圈表示一个城市,两个圆圈之间的连线表示对应城市之间的通信线路,连线上的数值表示该通信线路的造价。这一描述结构为图状结构,利用计算机可以求出满足要求的最小造价通信网络,如图 1.2 (b) 所示。如图 1.2 (a) 所示这种数据结构中,数据之间的关系是多对多的非线性关系,我们称这种数据结构为图形结构。

由以上三个例子可见,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,可以说数据结构课程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。

学习数据结构的目的是为了了解计算机处理对象的特性,将实际问题中所涉及的处理对象在计算机中表示出来并对它们进行处理。与此同时,通过算法训练来提高我们的思维能力,通过程序设计的技能训练来促进我们的综合应用能力和专业素质的提高。

1.1.2 学习数据结构的必要性

随着计算机运行速度的加快和存储(内存)容量的不断加大,有的人可能认为程序的运行效率变得越来越不重要了。然而,计算机的功能越强大,人们就越想去尝试解决更加复杂的问题。而更复杂的问题需要更大的计算量,这使得对高效率的程序的需求更加明显,工作越复杂就越偏离人们的日常经验。因此,学习数据结构,掌握各种前人设计的算法的运行效率并自己设计高效率的算法是非常必要的。

数据结构不仅是计算机专业教学计划中的核心课程之一,而且已逐步成为非计算机专业的主要选修课程之一。数据结构与数学、计算机硬件和软件的关系十分密切,是介于数学、计算机硬件和计算机软件之间的一门核心课程。在计算机科学中,数据结构不仅是一般非数值计算程序设计的基础,而且是设计和实现汇编语言、编译程序、操作系统、数据库系



统以及其他系统程序和大型应用程序的重要基础。打好数据结构课程的扎实基础,对于学习计算机专业其他课程,如编译原理、数据库系统原理、计算机网络基础等都十分有益。

1.2 数据的逻辑结构

1.2.1 基本概念

数据 (Data): 数据是信息的载体,它能够被计算机识别、存储和加工处理。它是计算机程序加工的原料,应用程序处理各种各样的数据。计算机科学中,数据的含义相当广泛,是指能被计算机加工处理的所有对象,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

数据元素 (Data Element): 数据元素是数据的基本单位,在计算机中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组成,也可以只由一个数据项组成,例如,表 1.1 “某班级学生学籍档案信息”中的一条记录、图 1.2 “用图描述通信网络问题”中的一个城市都可称为一个数据元素。表 1.1 中每个学生的学籍信息作为一个数据元素,在表中占一行,每个数据元素由学号、姓名、性别、年龄和入学成绩等 5 个数据项组成。数据元素又被称为元素、节点 (node)、顶点、记录 (record) 等。

数据项 (Data Item): 数据项是数据不可分割的、具有独立意义的最小数据单位,是对数据元素属性的描述。数据项也称为域或字段 (Field)。在表 1.1 中,每个数据元素由 5 个数据项组成。

数据类型 (Data Type): 数据类型是一组性质相同的值的集合以及定义在这个值的集合上的一组操作的总称。每个数据项属于某一确定的基本数据类型。如表 1.1 中,学号为数值型、姓名为字符型等。

数据对象 (Data Object): 数据对象是性质相同的数据元素的集合,是数据的一个子集。例如,整数数据对象是集合 $N=\{0, \pm 1, \pm 2, \dots\}$, 字母字符数据对象是集合 $C=\{'A', 'B', \dots, 'Z'\}$, 如表 1.1 所示的学籍表也可看做一个数据对象。由此可看出,不论数据元素集合是无限集 (如整数集)、有限集 (如字符集),还是由多个数据项组成的复合数据元素 (如表 1.1 “某班级学生学籍档案信息”),只要性质相同,就都是同一个数据对象。

数据结构 (Data Structure): 数据结构的基本含义是指数据元素之间的关系,它是按照某种关系组织起来的一批数据,以一定的存储方式把它们存储到计算机存储器中,并在这些数据上定义了一个运算的集合。在任何问题中,数据元素都不是孤立存在的,而是在它们之间存在着某种关系,数据元素之间的这种相互关系就称为结构,带有结构的数据对象称为数据结构。

1.2.2 逻辑结构的描述

1. 数据逻辑结构的基本分类

在任何一个问题中,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,数据的逻辑

结构通常划分成下面四种基本结构：

- (1) 集合：在集合结构中，数据元素间不存在任何关系，这是数据结构的一种特殊情况。集合是元素关系极为松散的一种结构，不在本书的讨论范围之内。
- (2) 线性结构：该数据结构中的数据元素之间存在着一对一的关系。
- (3) 树形结构：该数据结构中的数据元素之间存在着一对多的关系。
- (4) 图形结构：该数据结构中的数据元素之间存在着多对多的关系，图形结构也称做网状结构。

上述四种基本结构的关系如图 1.3 所示。

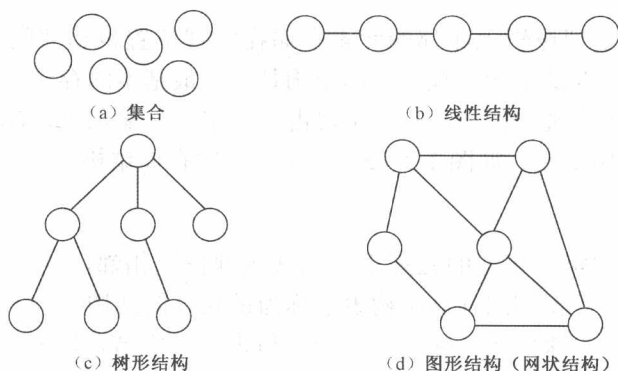


图 1.3 四种基本逻辑结构示意图

2. 数据逻辑结构的数学定义方法

下面用数学方法给出数据的逻辑结构定义。从上面所介绍的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 $S=(D,R)$ 的形式来表示。其中， D 是数据元素的有限集， R 是 D 上关系的有限集。

二元组 $S=(D,R)$ 中前驱和后继的关系可以描述如下：假设 a_1 、 a_2 是 D 中的两个元素，则在二元组 $\langle a_1, a_2 \rangle$ 中， a_1 是 a_2 的直接前驱， a_2 是 a_1 的直接后继。

例 1.4 用上面的数学方法给出一周 7 天的数据逻辑结构。设 a_1 、 a_2 、 a_3 、 a_4 、 a_5 、 a_6 、 a_7 分别表示星期一至星期日，这是线性结构。

$$S=(D, R)$$

$$D=\{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$$

$$R=\{\langle a_1, a_2 \rangle, \langle a_2, a_3 \rangle, \langle a_3, a_4 \rangle, \langle a_4, a_5 \rangle, \langle a_5, a_6 \rangle, \langle a_6, a_7 \rangle\}$$

该逻辑结构也可用图 1.4 表示。

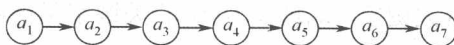


图 1.4 一周 7 天数据结构示意图

图中圆框表示一个节点，圆框内的符号是该节点的值，带箭头的线段表示前驱与后继的关系。



1.3 数据的存储结构

我们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表示称为存储结构, 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。数据的存储结构有:

1. 顺序存储

顺序存储方法就是把逻辑上相邻的元素存储在物理位置也相邻的存储单元中, 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。例如, 一个字母占一个字节, 输入 A、B、C、D、E 并存储在 1000 起始的连续的存储单元, 如图 1.5 (a) 所示为顺序存储结构。

2. 链式存储

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, 由此得到的存储结构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。例如, 一个字母占一个字节, 输入 A、B、C、D 以链式存储结构进行存储, 如图 1.5 (b) 所示为链式存储结构。

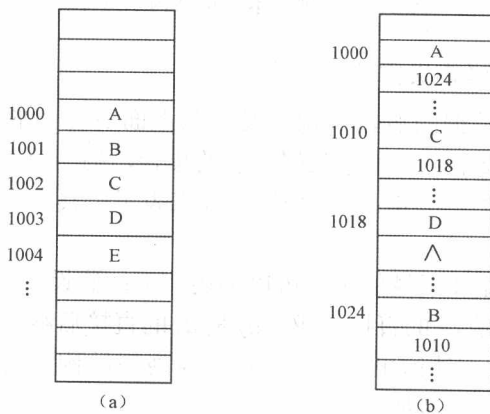


图 1.5 顺序存储和链式存储结构示意图

3. 索引存储

索引存储方法是指存放元素的同时, 还建立附加的索引表, 索引表中的每一项称为索引项, 索引项的一般形式是(关键字, 地址), 其中的关键字是能唯一标识一个节点的那些数据项。

4. 散列存储

散列存储是通过构造散列函数来确定数据存储地址或查找地址的。

1.4 算法和算法分析

1.4.1 算法特性

1. 算法 (Algorithm)

简单地说, 算法就是解决特定问题的方法。严格地说, 算法是由若干条指令组成的有穷序列, 其中每条指令表示计算机的一个或多个操作。例如, 将一组给定的数据由小到大进行排序, 解决的方法有若干种, 而每一种排序方法就是一种算法。

2. 算法的特性

一个算法必须具有以下五个特性:

- (1) 有穷性。一个算法必须在有穷步之后结束, 即必须在有限时间内完成, 不能形成无穷循环。
- (2) 确定性。算法中每条指令必须有确切的含义, 不会产生二义性。
- (3) 可行性。算法中描述的操作都是可以通过已经实现的基本运算执行有限次来实现的。
- (4) 输入。一个算法有零个或多个的输入, 这些输入取决于某个特定的对象的集合。
- (5) 输出。一个算法有一个或多个结果输出。

3. 算法与程序的区别

算法的含义与程序十分相似, 但又有区别。一个程序不一定满足有穷性。例如, 操作系统, 只要整个系统不遭破坏, 它将永远不会停止, 即使没有作业需要处理, 它仍处于动态等待中。因此, 操作系统不是一个算法。另一方面, 程序中的指令必须是机器可执行的, 而算法中的指令则无此限制。算法代表了对问题的解, 而程序则是算法在计算机上的特定的实现。一个算法若用程序设计语言来描述, 则它就是一个程序。

4. 一个好算法应该达到的目标

算法与数据结构是相辅相成的。解决某一特定类型问题的算法可以选定不同的数据结构, 而且选择恰当与否直接影响算法的效率。反之, 一种数据结构的优劣由各种算法的执行来体现。

设计一个好的算法可以从以下几个方面考虑:

- (1) 正确性。算法是为了针对解决具体问题而提出的, 算法的正确与否必须满足解决实际问题的需要, 要经得起一切可能的输入数据的考验。
- (2) 可读性。一个算法应当思路清晰、层次分明、简单明了、易读易懂。
- (3) 健壮性。当输入数据非法时, 算法应能适当地做出反应或进行处理, 不至引起严重后果。
- (4) 高效率。要求算法的执行时间要尽可能地短, 存储空间要尽可能地少, 即做到既节省时间又节省空间。