

◆ 高等教育应用型本科人才培养系列教材

数据结构

SHUJU JIEGOU

魏连锁 主编
李丽丽 副主编



高等教育应用型本科人才培养系列教材

数 据 结 构

魏连锁 主 编

李丽丽 副主编



哈尔滨工程大学出版社
Harbin Engineering University Press

内容简介

本书共 11 章,内容包括:绪论,线性表,栈和队列,串,数组、特殊矩阵和广义表,递归,树和森林,图,查找,排序,文件与外排序。本书内容符合数据结构课程要求,充分考虑了计算机专业学生的培养目标。在内容的组织上,由浅入深、循序渐进。书中采用 C 语言作为数据结构和算法的描述语言,在注重基本知识和基本概念的学习的同时,更注重学生能力的培养。

本书适合应用型本科人才培养的教学使用。

图书在版编目(CIP)数据

数据结构/魏连锁主编. —哈尔滨:哈尔滨工程大学出版社,2018.7

ISBN 978 - 7 - 5661 - 2016 - 8

I. ①数… II. ①魏… III. ①数据结构 - 高等学校 - 教材 IV. ①TP311.12

中国版本图书馆 CIP 数据核字(2018)第 150756 号

选题策划 夏飞洋

责任编辑 夏飞洋

封面设计 刘长友

出版发行 哈尔滨工程大学出版社
社 址 哈尔滨市南岗区南通大街 145 号
邮政编码 150001
发行电话 0451 - 82519328
传 真 0451 - 82519699
经 销 新华书店
印 刷 哈尔滨市石桥印务有限公司
开 本 787 mm × 1 092 mm 1/16
印 张 23.25
字 数 580 千字
版 次 2018 年 7 月第 1 版
印 次 2018 年 7 月第 1 次印刷
定 价 58.00 元

<http://www.hrbeupress.com>

E-mail:heupress@hrbeu.edu.cn

前 言

“数据结构”课程是计算机专业的基础课程和核心课程之一。其课程的核心内容是对数据的存储、组织和处理的问题进行研究。数据结构课程也是与实际紧密结合的理论课程,是进行算法设计、系统程序和应用程序设计的重要基础。而且本门课程的学习为学生后续学习数据库概论、操作系统、软件工程等软件课程提供了必要的知识基础,也为软件设计水平的提高打下了良好的基础。

本书是为“数据结构”课程所编写的教材,其内容符合数据结构课程要求。全书分为11章。

第1章绪论。介绍什么是数据结构,以及数据结构中常用的基本概念;阐述了数据结构所研究的问题与内容;对算法描述和分析进行了简要的介绍,为学习“数据结构”课程提供了必要的引导。

第2章线性表。介绍线性表的基本概念;线性表的两种存储结构;顺序表和单链表上实现基本运算的算法。

第3章栈和队列。介绍栈和队列的两种数据结构的定义、表示方法、基本操作,以及如何利用这两种数据结构解决实际问题。

第4章串。介绍串的定义及相关概念;串的定长顺序存储表示及基本操作;串的堆分配存储表示及基本操作;串的链式存储表示及基本操作;串的模式匹配算法。

第5章数组、特殊矩阵和广义表。介绍数组、特殊矩阵和广义表的定义;数组的顺序存储表示及实现;特殊矩阵(对称矩阵、三角矩阵和带状矩阵)的基本存储方式;特殊矩阵的存储表示,以及特殊矩阵的简单算法;广义表的性质、存储结构及其基本操作。

第6章递归。介绍递归的定义;递归算法的执行过程;递归算法的设计,以及利用递归算法解决实际问题;递归算法和非递归算法的区别;递归算法到非递归算法的转换。

第7章树和森林。介绍树形结构的基本概念和术语;二叉树的定义及其存储结构,二叉树的遍历概念和遍历算法;树和森林的定义、树的存储结构以及树、森林与二叉树之间的相互转换方法;构造哈夫曼树和设计哈夫曼编码的方法。

第8章图。介绍图的概念和相关术语;图的邻接矩阵表示法和邻接表表示法;连通图遍历的基本思想和算法;最小生成树的有关概念和算法;图的最短路径的有关概念和算法;拓扑排序的有关概念和算法。

第9章查找。介绍查找表的定义、分类和各类的特点;顺序查找和二分查找的思想和算法;二叉排序树的概念和有关运算的实现方法;哈希表、哈希函数的构造方法,以及处理冲突的方法;掌握散列存储和散列查找的基本思想及有关方法、算法。

第10章排序。介绍各种内部排序方法的指导思想和特点;几种内部排序算法和其基本思想;几种内部排序算法的优缺点、时空性能和适用场合。

第11章文件与外排序。介绍文件及其相关概念;文件的基本操作;文件的组织方式;外

排序的基本过程。

本书内容符合“数据结构”课程要求,充分考虑了计算机专业学生的培养目标。在内容的组织上,由浅入深、循序渐进。书中采用 C 语言作为数据结构和算法的描述语言,在注重基本知识和基本概念的学习的同时,更注重学生能力的培养。在教学过程中可以采用启发式案例教学的方法。

本书由齐齐哈尔大学魏连锁担任主编,齐齐哈尔大学李丽丽担任副主编。其中,第 1 章至第 4 章、第 8 章、第 10 章由魏连锁编写;第 5 章至第 7 章、第 9 章、第 11 章由李丽丽编写。魏连锁负责全书审阅。

限于作者水平,书中难免会有疏漏之处,恳请读者批评指正,以使本书得以改进和完善。

编 者

2018 年 4 月

目 录

第 1 章 绪论	1
1.1 数据结构基本概念和术语	1
1.2 数据类型	7
1.3 算法及其分析	9
习题 1	16
第 2 章 线性表	21
2.1 线性表定义和基本操作	21
2.2 线性表顺序存储结构	24
2.3 线性表链式存储结构	31
2.4 线性表的应用	48
习题 2	51
第 3 章 栈和队列	57
3.1 栈	57
3.2 队列	74
习题 3	89
第 4 章 串	95
4.1 串类型的定义	95
4.2 串的存储表示和实现	96
4.3 串的模式匹配算法	108
习题 4	112
第 5 章 数组、特殊矩阵和广义表	117
5.1 数组	117
5.2 特殊矩阵	123
5.3 稀疏矩阵	127
5.4 广义表	137
习题 5	149
第 6 章 递归	155
6.1 递归的定义	155
6.2 递归算法的执行过程	157
6.3 递归算法的设计	159
6.4 递归算法到非递归算法的转换	161
6.5 递归程序设计实例	165
习题 6	167

第7章 树和森林	169
7.1 树的基本概念	169
7.2 二叉树的定义与性质	178
7.3 二叉树的存储结构	184
7.4 二叉树的遍历	188
7.5 二叉树的基本算法及实现	194
7.6 线索二叉树	197
7.7 哈夫曼树	202
7.8 树和森林	209
习题7	213
第8章 图	222
8.1 图的基本概念	222
8.2 图的存储结构	226
8.3 图的遍历	234
8.4 图的连通性问题	236
8.5 最小生成树	239
8.6 最短路径	243
8.7 拓扑排序	248
8.8 AOE网与关键路径	250
习题8	253
第9章 查找	262
9.1 查找的基本概念	262
9.2 静态查找表	263
9.3 动态查找	269
9.4 哈希表查找	291
习题9	298
第10章 排序	307
10.1 排序的基本概念	307
10.2 插入排序	308
10.3 交换排序	312
10.4 选择排序	317
10.5 归并排序	321
10.6 基数排序	323
10.7 各种内排序方法的比较和选择	325
习题10	327
第11章 文件与外排序	334
11.1 基本概念	334
11.2 文件的组织方式	336
11.3 外排序的基本过程	343
习题11	344

参考文献	346
数据结构考试大纲	347
I 课程性质与课程目标	347
II 考核目标	348
III 课程内容与考核要求	348
IV 关于大纲的说明与考核实施要求	359
题型举例	361

第 1 章 绪 论

自 1946 年第一台计算机 ENIAC 问世以来,人类进入了现代电子数字计算机的时代。随着计算机技术的不断发展,计算机科学也随之产生。如何有效地组织存储数据,如何有效地处理数据成为计算机科学的基本研究内容。

本章主要介绍了什么是数据结构、数据结构中常用的基本概念及数据结构所研究的问题与内容,并对算法描述和分析进行了简要的介绍。通过本章的学习,使读者对数据结构课程有一个初步的了解,并为后续内容的学习提供参考。

1.1 数据结构基本概念和术语

在计算机中,人们通常会以数据的形式来表示输入、输出以及中间结果。程序的执行效率受到不同计算模型以及不同平台环境的影响,对数据的存储、组织、转换等操作会直接影响算法效率。

数据结构是指相互之间存在一种或多种特定关系的数据元素的集合。“数据结构”在计算机科学中是一门综合性的专业基础课,数据结构是介于数学、计算机硬件、软件三者之间的一门核心课程。数据结构这一门课的内容不仅是一般程序设计的基础,而且还是设计和实现编译程序、操作系统、数据库系统及其他系统程序的重要基础,它主要研究在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和操作。因此数据结构是以“数据”的信息的表现形式为研究对象,并研究支持高效算法的数据信息处理策略、技巧和方法。

从 1968 年开始,“数据结构”作为一门独立的课程开始设立。1968 年,美国唐纳德·克努特教授开创了数据结构的最初体系,他所著的《计算机程序设计艺术》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。在此之前,虽然数据结构的一些内容在其他课程中有所体现,并且在美国一些大学中也设置了数据结构这门课程,但是对其范围没有做出十分明确的规定。

计算机科学是一门研究用计算机进行信息表示和处理的科学。这里涉及两个问题,即信息的表示和信息的信息的处理。通常情况下,用计算机解决实际问题需要经过以下几个阶段:

(1)分析问题阶段。在这个阶段中,首先对计算机要解决的问题进行分析。即明确问题是什么,要解决什么问题,用户的需求是什么,已知条件是什么,如何根据已知的条件进行处理,进而达到最终目的。

(2) 建立模型及设计算法阶段。算法是解决问题方案的准确而完整的描述,是一系列清晰指令的集合。算法代表着用系统的方法描述解决问题的策略机制。正确的算法可以实现规范的输入,并在有限时间内输出所需要的结果。设计算法阶段需要寻找解决问题的策略和方法,并且使用适当的工具加以详细的描述和说明。如果一个算法有缺陷,或不适合于某个问题,执行这个算法将不会解决这个问题。不同的算法可能用不同的时间、空间或效率来完成同样的任务。一个算法的优劣可以用空间复杂度与时间复杂度来衡量。首先,建立模型。通过对实际问题的多次抽象,建立能被计算机存储和处理的数据模型。其次,定义数据。利用恰当的表达工具定义数据的逻辑结构和物理结构。逻辑结构的定义是对数据模型的抽象逻辑描述。逻辑结构独立于计算机,可以不考虑机器实现,只需要考虑要解决问题的抽象数据本身和其逻辑关系的表达。物理结构的定义则正好相反,需要将数据和其关系映射到计算机内,转化成计算机能够存储和处理的形式。最后,寻找正确的算法。找到要解决问题的方法,并利用适当的工具加以详细描述。在算法设计阶段,由于许多问题没有可用的算法,所以需要程序设计人员自己去探索、发现或创新。

(3) 编写程序及调试运行阶段。设计适当的算法之后,就需要使用计算机程序设计语言进行编写程序。编程的语言包括汇编语言、机器语言和高级语言。常用的高级语言有 C 语言和 Java 语言。编写程序就是将第二阶段所设计的算法翻译成计算机能够理解和执行的语言,并得到最终的相应结果。

1.1.1 数据结构的基本概念

1. 数据

数据就是指能够被计算机所接收和处理的对象。那些能够被输入到计算机中,并且能够被计算机处理的各种符号,都可以称为数据。随着计算机技术不断的发展变化,数据的形式也越来越多样化,除整数、小数等表示数字的数据外,字符、图形、图像、表格等能够被计算机存储和处理的对象也可以称之为数据。总之,数据已经成为信息的载体,而信息是数据的内涵。

2. 数据元素

数据元素是数据的基本单位。通常情况下,数据元素在程序设计过程中是作为一个整体进行处理的,一个数据元素就是数据中的一个“个体”。数据元素通常具有完整而确定的实际意义,表示同一个对象、多个特征。在不同的情况下,数据元素也可以被称为元素、顶点或者是记录。数据元素是作为一个完整的数据进行存储和处理的。

3. 数据项

如果说数据元素是数据的基本单位的话,数据项就是数据不可分割的最小单位。一个数据元素是由若干个数据项组成的。数据元素是数据项的集合。数据项就是具有独立逻辑意义且不可再分割的数据。数据项也被称为字段或域。如图书信息中图书名、作者、出版社等就是数据项,而书目信息作为一个数据元素由图书名、作者、出版社等数据项构成。从逻辑角度来看,一个数据项能够反映其描述对象的一方面的属性,而数据元素则由多个数据项共同描述对象的多个特征,是一个整体。从存储角度来看,数据元素作为基本单位,整个数据的存取和处理都是以数据元素为单位进行操作的。

例如,表 1-1 所示的为学生信息统计表。表中每个学生信息即为一条记录,每个记录由姓名、性别、年龄、出生日期、班级以及成绩几个数据项组成。其中姓名、性别、年龄、出生

日期、班级为原子项；成绩为组合项，分为数学、语文、英语、物理4个原子项。

表 1-1 学生信息统计表

姓名	性别	年龄 (周岁)	出生日期	班级	成绩			
					数学	语文	英语	物理
张臣	男	21	19970313	141	90	80	75	80
李月	女	19	19990408	141	95	78	85	67
王晨	女	20	19980526	142	96	74	82	80
.....

4. 数据对象

数据对象是由具有相同性质的数据元素构成的集合。数据对象就是一个集合，而数据元素则是一个个体。数据元素描述的是一个客观对象，具有共同特征的对象组合在一起，即得到一组数据元素，成为一个集合，这个集合就是一个数据对象。通常情况下，在计算机中，一个数据对象会被组织成一个文件，并以文件的形式存储在计算机系统中，作为整体进行存储和维护。所以，数据对象就是数据存储管理单位。

数据项、数据元素、数据对象之间的关系：一个数据元素是由若干相关的数据项构成的，一个数据对象是由相关的数据元素构成的。它们三者之间是被包含与包含的关系。数据项、数据元素、数据对象是数据逻辑结构的层次单位，也是存储结构的层次单位。

5. 数据结构

数据结构是数据对象以及定义在这组数据对象上的某种特定的关系。它是相互之间存在一种或多种特定关系的数据元素的集合。数据结构就是计算机存储和组织数据的一种方式。通常情况下，数据结构会影响数据元素的检索和存储效率。数据结构与其索引技术有关。数据结构可以形象地表示为：数据结构 = 数据元素的集合 + 一组关系的集合。

数据结构是一个二元组，其形式表示为

$$\text{Data_Structure} = (D, R)$$

式中 D ——数据元素的有限集合；

R ——在 D 上的关系的有限集合。

例如： $L = \{A, B, C, D, E\}$, $R = \{<\}$, $D_S = \{L, R\}$ 即是一个数据结构。

1.1.2 逻辑结构类型

研究数据结构不仅是研究数据的逻辑结构，还研究数据的物理结构。逻辑结构表示的是数据元素之间的逻辑关系。根据数据元素之间的关系的不同，数据的逻辑结构通常分为4种。

1. 集合结构

数据元素之间没有任何的联系，只是同属于同一个集合，如图 1-1 所示。

2. 线性结构

数据元素之间存在着一对一的关系。线性结构也称为线性表，如图 1-2 所示。

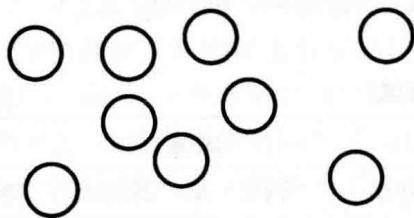


图 1-1 集合结构



图 1-2 线性结构

3. 树形结构

数据元素之间存在着一对多的关系。树形结构有良好的层次性,所以表示具有层次结构的数据时,树形结构是一种很好的选择,如图 1-3 所示。

4. 图状结构

图状结构又称图,其数据元素之间存在着多对多的关系。这种结构描述数据元素之间存在的复杂的网络关系,如图 1-4 所示。

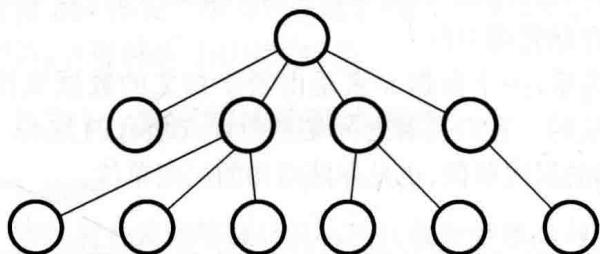


图 1-3 树形结构

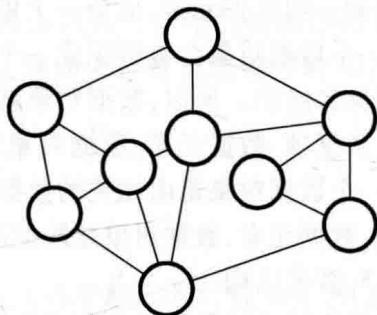


图 1-4 图状结构

以下 3 个例子说明 3 种不同的数据结构。

【例 1-1】 如表 1-2 所示,在数据库管理系统中,通常用记录来表示表中的一个数据元素。可以看出,图书信息表是一个线性的数据结构,表中的每行表示一个记录。每个数据元素由 ISBN 号、教材名称、主编、出版社、出版时间、估价等数据项组成,数据元素之间存在着一对一的关系。

表 1-2 图书信息表

ISBN 号	教材名称	主编	出版社	出版时间	估价 /元
9787040406641	数据库系统概论	王珊	高等教育出版社	2014.09	39.6
9787302330981	软件工程导论(第 6 版)	张海藩	清华大学出版社	2013.10	39.5
9787121225550	网络舆情分析师教程	薛大龙	电子工业出版社	2014.04	26
.....

【例 1-2】 Linux 文件目录结构是一个典型的树形结构。如图 1-5 所示,“/”为 Linux 系统的根目录,是整个树形结构的根节点;etc、usr、var、home 等都是“/”的下一级目录。在

树形结构中通常会把一个节点的直接前驱称为该节点的父节点,一个节点的直接后继称为该节点的子节点。因此根节点“/”为“etc”等节点的父节点,而“etc”等节点称为“/”的子节点。通过观察可以发现,与线性结构不同,树形结构中的节点是一对多的关系,即一个父节点可以对应多个子节点。树形结构也是一种严格的层次结构,上层节点与下层之间具有层次关系,而且在这种关系中,根节点作为整个树形结构的起始节点。树形结构的这种层次关系是不能颠倒的,即只能先有父节点,才能再有子节点,不存在只有子节点,没有父节点的情况。

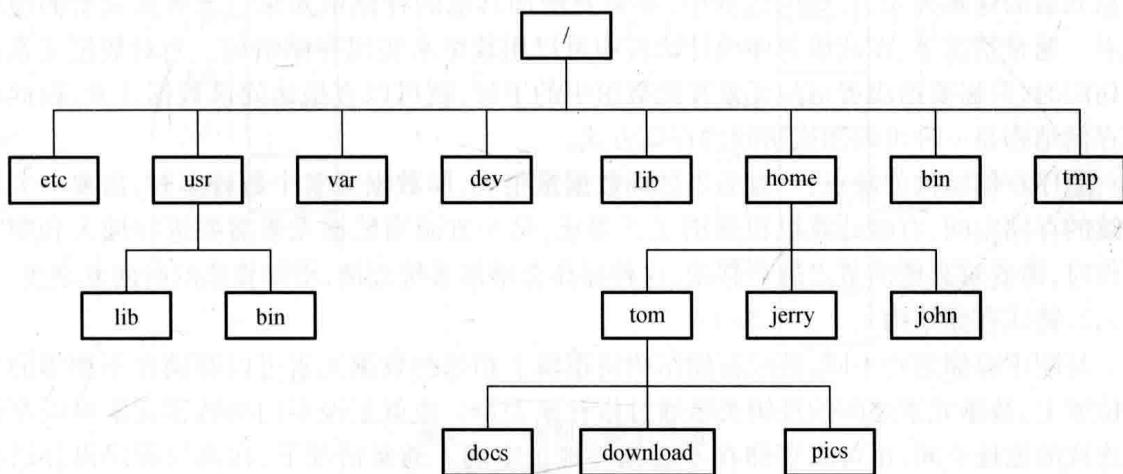


图 1-5 Linux 文件目录结构

【例 1-3】 图 1-6 为城市交通图。由图可知,这是个典型的图状结构。图中每个城市就是一个数据元素,在图状结构中,这样的数据元素称为顶点。不难发现,各顶点之间存在着多对多的关系。有些顶点和顶点之间存在边,即具有连通性。图状结构也可以称为网状结构。

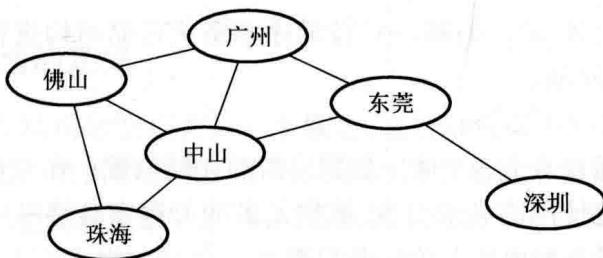


图 1-6 城市交通图

1.1.3 数据的存储结构

在数据结构中,逻辑结构通常被用来描述数据元素之间存在的逻辑关系。上述 4 种关系均可以用二元组来表示。逻辑结构是为了研究所要解决的问题。在计算机中除了用逻辑结构来描述这些数据元素之间的关系外,还需要研究这些数据元素在计算机中的表示,即如何在计算机中操作。数据结构在计算机中的存储和表示,称为数据的物理结构。

在计算机中,数据的存储结构分成顺序存储结构和链式存储结构两类。常用以下 4 种

存储方法。

1. 顺序存储结构

顺序存储结构经常事先约定好数据元素的先后次序,然后将这些数据元素逐一存放在连续的物理存储空间上。也就是把逻辑上相邻的节点存储在物理位置也相邻的存储单元中,数据元素的物理位置的相邻关系表示数据元素逻辑上的相邻关系。

顺序存储结构是一种简单且易于实现的存储方式。因为在采用顺序存储结构存储数据元素时,数据元素的访问,只需要知道数据元素在物理存储空间的位置关系,就能够得到数据元素的逻辑关系,存储的过程中,不需要增加其他的存储单元来记录数据元素的逻辑关系。通常情况下,在高级程序设计语言中可以用数组来实现存储结构。当对数据元素进行访问时,只需要给出要访问元素在此数组中的下标,就可以直接访问该数据元素,因此顺序存储结构是一种可以随机访问的存储方式。

顺序存储结构的缺点:一方面当处理数据量很大,即数据元素个数较多时,需要一大块连续的存储空间,有时计算机很难满足其要求;另一方面当数据元素需要进行插入和删除操作时,需要对其他的节点进行移动,这种操作会增加系统开销,增加算法的时间复杂度。

2. 链式存储结构

与顺序存储结构不同,链式存储结构将逻辑上相邻的数据元素可以存储在不相邻的存储位置上,数据元素之间的逻辑关系通过指针来表示。也就是说不同的数据元素可以存储在连续的地址空间,也可以存储在不连续的地址空间。通常情况下,在高级程序设计语言中可以用指针类型来实现链式存储中逻辑关系的描述。

链式存储结构的存储空间利用率较低,因为除了存储数据元素的这部分存储空间外,存储空间还有一部分用来存储数据元素之间的逻辑关系,即指针。另外,链式存储结构不能实现随机访问,因为逻辑关系相邻的元素不一定存储在物理位置相邻的存储空间。链式存储结构具有容易修改的特点,由于数据元素的逻辑关系是通过指针来描述的,因此在插入、删除数据元素的操作时,不需要像顺序存储结构一样对数据元素进行移动,只需要对相关指针进行修改。

【例 1-4】 序列 {20,35,16,38,56} 按顺序和链式存储结构进行存储,其在机器内存储的示意图如图 1-7 所示。

3. 索引存储结构

索引存储结构是通过索引表来确定数据元素的存储位置。在存储数据元素时,将数据元素的关键字和存储地址构造成索引表,数据元素的关键字能够唯一标识数据元素,而通过存储地址可以访问其存储地址上的数据元素。

索引存储结构也能够实现对数据元素的随机访问。当需要对数据进行插入、删除等操作时,可以不移动存储空间上的数据元素,只需要修改和移动索引表中的相关节点的存储地址信息。

索引存储结构可以分为稀疏索引和稠密索引两种:稀疏索引中一组数据元素对应一个索引项;稠密索引中每一个数据元素对应一个索引项。

4. 散列存储结构

散列存储结构是将数据元素的关键字转换成其存储地址,这种结构也称为哈希存储结构。这种结构适用于高速检索,效率近似于随机存取。

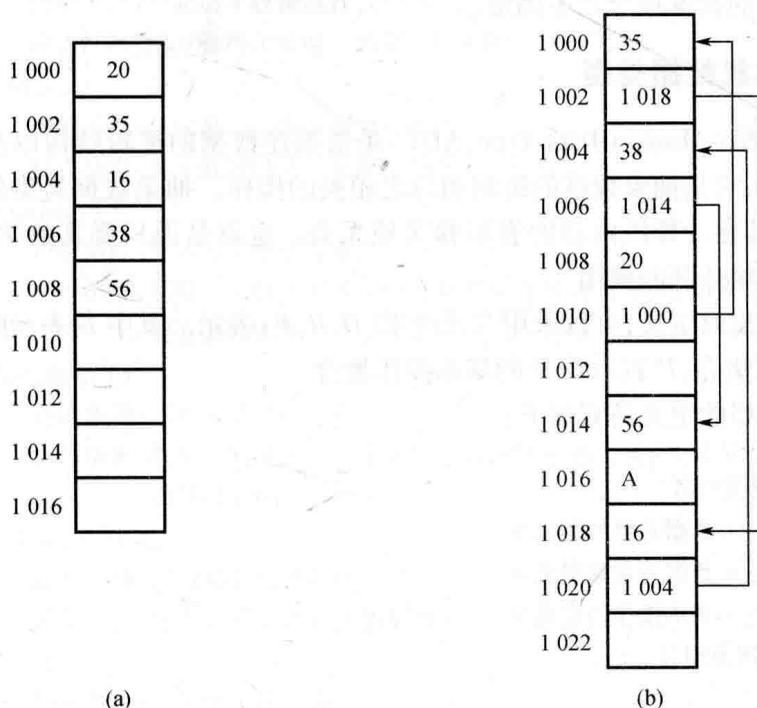


图 1-7 序列存储示意图

(a) 顺序存储; (b) 链式存储

4 种存储结构可以单独使用,也可以结合起来使用。相同的逻辑结构可以采用不同的存储结构,通过不同的存储结构来满足不同应用的具体要求。

1.2 数据类型

1.2.1 数据类型的定义

数据类型是与数据结构紧密相关的一个概念,它是对数据类型的进一步抽象,是基本数据类型的延伸和发展。数据类型是一组值的集合以及在这组值上定义的操作的总称。也就是说数据类型不只包括其数据对象,也包括在数据对象上的操作。在高级程序设计语言中,通常可以用数据类型来描述数据对象的特性。数据类型包括两方面的含义:一方面它是值的集合,其中的数据具有相同的类型;另一方面,它也是一组操作集合,这组操作是作用在值的集合上的。在高级语言中定义的常量、变量等都有其不同的数据类型。如整型、布尔类型等,对于整数类型,是除了包含其整数值取值范围的所有数据的集合外,还包含定义在其值上的加减乘除等运算。

通常情况下,可以将数据类型按其“值”的特性划分成原子类型和结构类型。

①原子类型,其值不可分解,如整型、布尔类型。

②结构类型,其值由若干个分量组成,通常情况下,这些分量由用户自己定义,如数组、结构体等类型。在结构类型中又可将数据类型进一步细分成静态聚合类型和动态聚合类型。静态聚合类型的变量的值由固定个数的数据项组成复合数据类型;动态聚合类型中组

成复合数据类型的数据项个数不固定。

1.2.2 抽象数据类型

抽象数据类型(Abstract Data Type, ADT)是指描述数据的逻辑结构以及在这些逻辑结构上的一组操作,它是抽象数据的组织和与之相关的操作。抽象数据类型的定义取决于它的逻辑特性,与其在计算机内部的表示和实现无关。也就是说只要其数学特性不变,其内部的变化不会影响外部的使用。

抽象数据类型的定义,可以采用三元组来(D,R,P)表示。其中 D 表示的是数据对象, R 表示 D 上的关系集合, P 表示对 D 的基本操作集合。

抽象数据类型的定义格式如下:

```
ADT 抽象数据类型名 {
    数据对象: <数据对象的定义>
    数据关系: <数据关系的定义>
    基本操作: <基本操作的定义>
}
```

ADT 抽象数据类型名

其中,可以用伪代码来定义数据对象和数据关系。定义基本操作格式如下:

基本操作名:

```
    初始条件: <初始条件描述>
    操作结果: <操作结果描述>
```

抽象数据类型和数据类型实质上是同一个概念。抽象数据类型是用户根据实际应用自定义的数据类型,不再局限于高级计算机语言中已有的数据类型。为了提高软件的利用率,软件系统的框架不再是建立在操作之上,而是建立在数据之上。这种程序设计方法,将在软件系统的每个相对的独立模块中,定义一组数据和在这组数据上的一组操作,并在模块的内部给出数据的表示和实现细节,在模块的外部使用的只是抽象的数据和抽象的操作。所以不难发现,当定义的数据类型的抽象层次越高,那么包含这个抽象数据类型的模块的利用率也就越高。

定义一个三元组抽象数据类型。

ADT Triplet

```
{
    数据对象 D:  $D = \{e_1, e_2, e_3 \mid e_1, e_2, e_3 \text{ 属于定义了关系的某个集合}\}$ 
    数据关系 R:  $R = \{ \langle e_1, e_2 \rangle, \langle e_2, e_3 \rangle \}$ 
    基本操作 P:
```

```
    InitTriplet(&T, v1, v2, v3) ;
```

操作结果:构造三元 T,元素 e_1, e_2 和 e_3 赋予参数 v_1, v_2, v_3 的值。

```
    DestroyTriplet(&T)
```

初始条件:三元组 T 已经存在。

操作结果:销毁三元组 T。

```
    Get(T, i, &e)
```

初始条件:三元组 T 已经存在, $1 \leq i \leq 3$,
操作结果:用 e 返回三元组 T 的第 i 个元素。

Put(&T,i,e)

初始条件:三元组 T 已经存在, $1 \leq i \leq 3$,
操作结果:将 e 值赋值给三元组 T 的第 i 个元素。

IsAscending(T)

初始条件:三元组 T 已经存在。
操作结果:判断三元组 T 的三个元素是否按升序进行排列,如果是,则返回 TRUE;否则返回 FALSE

IsDescending(T)

初始条件:三元组 T 已经存在。
操作结果:判断三元组 T 的三个元素是否按降序进行排列,如果是,则返回 TRUE;否则返回 FALSE

Max(T,&e)

初始条件:三元组 T 已经存在。
操作结果:将三元组 T 的最大值赋给 e。

Min(T,&e)

初始条件:三元组 T 已经存在。
操作结果:将三元组 T 的最小值赋给 e。

} ADT Triplet

基本运算中有两种参数,分别是赋值参数和引用参数。赋值参数用于为运算提供输入值;引用参数以 & 开头,可以提供输入值及用于返回运算结果。

抽象数据类型强调的是数据的抽象和数据封装。也就是说,抽象数据类型的每个操作都尽可能单一明确。这样可以减少操作功能上的重复。采用抽象数据类型可以隐藏内部的实现细节,做到实体的外部特性与内部实现细节分离。

1.3 算法及其分析

1.3.1 算法的概念及其基本特征

1. 算法的基本概念

在用计算机解决问题的过程中,需要通过程序设计语言编程实现。而算法就是描述解决问题方法的操作步骤的集合,是为了解决一个或者一类问题的有限的、确定的操作序列。程序、数据结构以及算法具有极密切的关系。在程序设计的过程中,需要先确定数据结构,然后根据所确定的数据结构来设计相应的操作步骤——算法。也就是说我们可以把数据结构、算法和程序表示成如下的关系:

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

解决任何一个问题都需要事先做出计划,并制定解决问题的步骤,而利用计算机解决问题,就是在所定义的数据结构上进行的相应操作。