



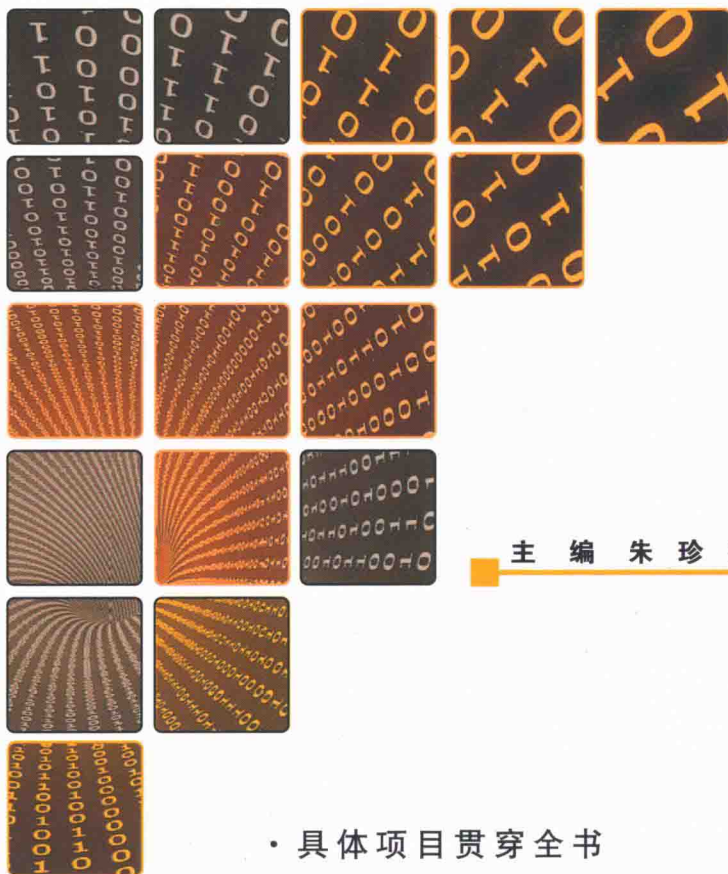
高职高专计算机
系列规划教材

PUP6

高职高专计算机系列规划教材

全国高职高专计算机 **立体化** 系列规划教材

数据结构及应用



主 编 朱 珍 徐丽新

- 具体项目贯穿全书
- 生动体现知识点的应用
- 书中算法均已上机测试



北京大学出版社
PEKING UNIVERSITY PRESS

全国高职高专计算机立体化系列规划教材

数据结构及应用

主 编	朱 珍	徐丽新	
副主编	黄 玲	李天祥	魏瑞娟
参 编	钟祥睿	卢世军	杨 咏



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

本书内容全面,包括认识数据结构、线性表及应用、栈及应用、队列及应用、字符串及应用、树及应用、图及应用、查找、内部排序、课程设计 10 个部分。为了便于读者学习,在讲解每一个知识点时都引入具体的项目,并结合图例进行分析,然后是算法描述,最后是具体程序实现。每一个例子都比较典型且知识点覆盖完整。本书所有算法都是用 C 语言描述,在 Visual C++ 6.0 环境下测试通过,能够直接运行。

本书可作为大中专院校的计算机相关专业数据结构的教材,也可作为计算机软件开发、考研和软件等级考试相关人员的参考书。

图书在版编目(CIP)数据

数据结构及应用/朱珍,徐丽新主编. —北京:北京大学出版社,2012.1

(全国高职高专计算机立体化系列规划教材)

ISBN 978-7-301-19801-8

I. ①数… II. ①朱…②徐… III. ①数据结构—高等职业教育—教材 IV. ①TP311.12

中国版本图书馆 CIP 数据核字(2011)第 240640 号

书 名: 数据结构及应用

著作责任者: 朱 珍 徐丽新 主编

责任编辑: 李彦红

标准书号: ISBN 978-7-301-19801-8/TP · 1197

出版者: 北京大学出版社

地 址: 北京市海淀区成府路 205 号 100871

网 址: <http://www.pup.cn> <http://www.pup6.cn>

电 话: 邮购部 62752015 发行部 62750672 编辑部 62750667 出版部 62754962

电子邮箱: pup_6@163.com

印刷者: 北京鑫海金澳胶印有限公司

发 行 者: 北京大学出版社

经 销 者: 新华书店

787mm×1092mm 16 开本 14.5 印张 336 千字

2012 年 1 月第 1 版 2012 年 1 月第 1 次印刷

定 价: 28.00 元

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

举报电话: 010-62752024

电子邮箱: fd@pup.pku.edu.cn

前 言

“数据结构及应用”是计算机类相关专业的专业基础和核心课程，是算法设计的基础。“数据结构及应用”就是完成将一个现实生活中的具体问题进行抽象表示的任务，研究如何将数据进行逻辑表示，再根据数据的逻辑结构表示为相应的存储结构，然后找到一个有效的解决问题的方法即算法，最后通过计算机程序设计语言编制程序、运行程序并得到最终的结果。

本书具有以下特点。

(1) 本书覆盖了数据结构中线性表、树和图的所有知识点，每种数据结构都使用了逻辑结构和存储结构进行描述，并对算法的实现尽可能采用多种实现方式，从而使读者对算法的理解更加深刻。

(2) 本书将每种数据结构用一个大项目贯穿，用项目、任务、子任务、模块划分各知识点，易于读者理解。在知识点的讲解过程中，循序渐进，由浅入深，先引出概念，再用例子说明，然后是算法描述，最后是具体程序实现。这样的层次易于读者理解和消化。

(3) 本书中各项目都配有实训。每个实训均包含两类项目，第一类项目的重点在于对所学知识的理解和应用，第二类项目的重点在于对实际问题的解决。通过这两个实训项目，不仅能帮助学生加深对基础理论知识的理解，也培养了学生解决实际应用问题的能力。

(4) 本书中各项目均配有小结和习题，方便学生课后复习、巩固。

(5) 为方便学生学习，本书的算法部分均采用 C 语言描述，实训项目也是完整的 C 语言程序，读者可以很方便地对书中的算法进行上机测试。

本书的内容主要分为 5 部分：第 1 部分是基础篇，包括项目 1，主要内容是数据结构概述和算法知识概述；第 2 部分是线性数据结构，包括项目 2、项目 3、项目 4 和项目 5，主要内容是线性表、栈、队列、字符串及其应用；第 3 部分是非线性数据结构，包括项目 6 和项目 7，主要内容是树和图及其应用；第 4 部分是查找和排序及其应用，包括项目 8 和项目 9；第 5 部分是课程设计。

在第 2、3、4 部分中，分别贯穿一个完整的项目，如第 2 部分的学生成绩管理系统、数制转换系统、学生答疑系统和文本编辑器，第 3 部分的哈弗曼编译器，第 4 部分的旅游景区管理信息系统，便于读者更好地学习和理解相关知识点。

本书的教学时数以 60~80 学时为宜，其中上机安排 40 学时左右。教师可以根据实际的教学时数和学生情况等自行调整教学进度和教学内容。

本书由广东工程职业技术学院朱珍、徐丽新担任主编，广东工程职业技术学院黄玲、四川科技职业学院李天祥和山东冶金技术学院魏瑞娟担任副主编，广东工程职业技术学院钟祥睿、卢世军、杨咏参与了编写。其中，朱珍编写了项目 2，徐丽新编写了项目 3 和项目 4，黄玲编写了项目 8 和项目 9，钟祥睿编写了项目 1 和项目 7，卢世军编写了项目 5，杨咏编写了项目 6，李天祥编写了课程设计，魏瑞娟审阅了全书。广东工程职业技术学院计算机信息系的领导和其他老师在本书编写过程中给予了大力的协助和支持，在此表示衷心的感谢。

由于时间紧迫，加之编者水平有限，书中难免存在疏漏之处，欢迎读者批评指正。

编 者
2011 年 5 月

目 录

项目 1 认识数据结构.....1	小结.....77
任务 1.1 了解数据结构研究的 主要内容.....2	实训：队列及应用.....78
任务 1.2 理解相关基本概念和术语.....4	习题.....78
任务 1.3 算法.....6	项目 5 字符串及应用
小结.....10	——文本编辑器.....80
实训：算法时间复杂度分析.....10	任务 5.1 理解字符串的逻辑结构.....81
习题.....11	任务 5.2 字符串的表示和实现.....84
项目 2 线性表及应用	任务 5.3 字符串的模式匹配算法.....92
——学生成绩管理系统.....14	任务 5.4 文本编辑器的实现.....98
任务 2.1 理解线性表的逻辑结构.....15	小结.....101
任务 2.2 线性表的顺序表示和实现.....18	实训：字符串及应用.....101
任务 2.3 线性表的链式表示和实现 ——学生成绩管理系统链表 实现.....29	习题.....102
任务 2.4 线性表应用举例.....40	项目 6 树及应用
小结.....41	——哈弗曼译码器.....103
实训：线性表.....42	任务 6.1 理解树的逻辑结构.....104
习题.....43	任务 6.2 二叉树的存储结构和基本 操作.....107
项目 3 栈及应用	任务 6.3 二叉树的遍历和线索化.....109
——数制转换系统.....45	任务 6.4 树和二叉树的转换.....118
任务 3.1 理解栈的逻辑结构.....46	任务 6.5 哈弗曼树及其应用.....122
任务 3.2 栈的顺序表示和实现.....47	小结.....126
任务 3.3 栈的链式表示和实现.....54	实训：二叉排序树的实现.....126
小结.....59	习题.....127
实训：栈及应用.....59	项目 7 图及应用
习题.....60	——旅游景区管理信息系统.....128
项目 4 队列及应用	任务 7.1 理解图的基本概念.....129
——学生答疑系统.....62	任务 7.2 图的存储结构——旅游景区 管理信息系统的物理实现.....132
任务 4.1 理解队列的逻辑结构.....63	任务 7.3 图的遍历.....137
任务 4.2 队列的顺序表示和实现.....65	任务 7.4 最小生成树.....140
任务 4.3 队列的链式表示和实现.....74	任务 7.5 最短路径.....143

任务 7.6 拓扑排序和关键路径	146	项目 9 内部排序	189
任务 7.7 旅游景区管理信息系统的 实现	152	任务 9.1 理解排序	190
小结	158	任务 9.2 学习插入排序	193
实训：图及应用	158	任务 9.3 学习交换排序	198
习题	159	任务 9.4 学习选择排序	205
项目 8 查找	163	任务 9.5 学生成绩管理系统排序案例	207
任务 8.1 理解查找	164	小结	216
任务 8.2 掌握基于线性表的查找	165	实训：排序	216
任务 8.3 掌握基于树的查找	177	习题	217
小结	186	课程设计	219
实训：查找	186	参考文献	223
习题	187		

项目 1

认识数据结构



教学目标

本项目将介绍数据结构的基本知识,包括其基本概念与术语、数据的逻辑结构与存储结构、算法描述与分析。通过本项目的学习,应了解数据结构的概念、算法的基本功能特征和算法的评价标准,掌握估算算法的时间复杂度的方法。



教学要求

知识要点	能力要求	相关知识
基本概念	掌握和理解常用数据结构的概念和术语	高级语言数据类型
算法描述	理解算法五要素的含义	
算法评价标准	理解和掌握算法的 4 个评价标准,掌握时间复杂度的求法	



引言

随着计算机技术的快速发展与日益普及,计算机应用的范围也越来越广泛。从最初的数值计算,发展到现在的数据处理、自动控制、信息处理、办公自动化等非数值计算领域。所处理的数据也从简单的数字发展到复杂的文字、图形、图像、音频和视频等数据。因此,要想高效地处理好这些数据,必须解决好 3 个方面的问题:第一,数据本身的特性及它们之间的关系;第二,如何有效地将数据组织存储在计算机内;第三,对于存储在计算机中的数据可以进行哪些操作,如何实现这些操作,对同一问题的不同操作方法如何进行评价。这些问题就是数据结构所要研究的主要内容。

本项目主要介绍数据结构的基本概念,数据的逻辑结构、存储结构及其关系,然后讨论算法的基本特征、评价标准及算法的时间复杂度的估算方法。

任务 1.1 了解数据结构研究的主要内容

【工作任务】

在理解数据结构的概念之前，应先了解数据结构研究的主要内容及范围。

在介绍数据结构定义之前，先来看看几个具体问题。

【例 1-1】公司员工信息管理。

某公司有“张兵”、“杨明”、“李婷婷”等员工。现公司想要用计算机管理其员工信息，要求能够完成以下操作。

- (1) 当有新员工时，能将新员工信息添加进来。
- (2) 当有员工辞职时，能够删除该员工信息。
- (3) 可以修改员工信息。
- (4) 能够以某种方式查找员工信息。

分析：

通过对以上问题的描述，可以把公司员工信息用表 1-1 表示出来。其中，每个员工的信息由员工号、姓名、性别、年龄、住址、电话、所属部门等组成，员工信息按一定的顺序线性排列，这就是解决该问题的模型(线性表)。有了模型以后，就可以围绕该模型设计算法，即实现员工信息的添加、修改、删除、查找等操作。

表 1-1 员工信息表

员工号	姓名	性别	年龄	住址	电话	所属部门
01001	张兵	男	42	中山路 12 号	3872	办公室
02001	杨明	男	38	解放路 35 号	5077	财务部
02002	李婷婷	女	29	北京路 5 号	5078	财务部
03001	王丽丽	女	21	环市路 189 号	2250	销售部
...	...	...	...	...	...	...

类似的还有学生信息管理系统、图书管理系统、飞机订票系统等，它们具有共同之处，即被处理的对象之间具有线性关系，这就是一类数据结构——线性结构。

【例 1-2】CBA 季后赛对阵形势。

在中国男子篮球职业联赛的每个赛季，进入季后赛的有 8 支球队，进行淘汰赛，胜者进入下一轮。现在希望得到各队对阵形势，以及输赢情况。

分析：

通过对以上问题的描述，可以把各队对阵形势用图 1.1 表示出来。这是一棵倒长的树，树根在最上面，树叶在下面。树根就是总冠军，树叶就是参赛的各支球队。内部的树杈和结点表示两支球隊对阵情况和获胜一方。如果要查看广东东莞银行队的比赛情况，可以从树根开始沿树杈到达表示广东东莞银行队的树叶即可。树也是一种数据结构，也能表达某些非数值计算问题。

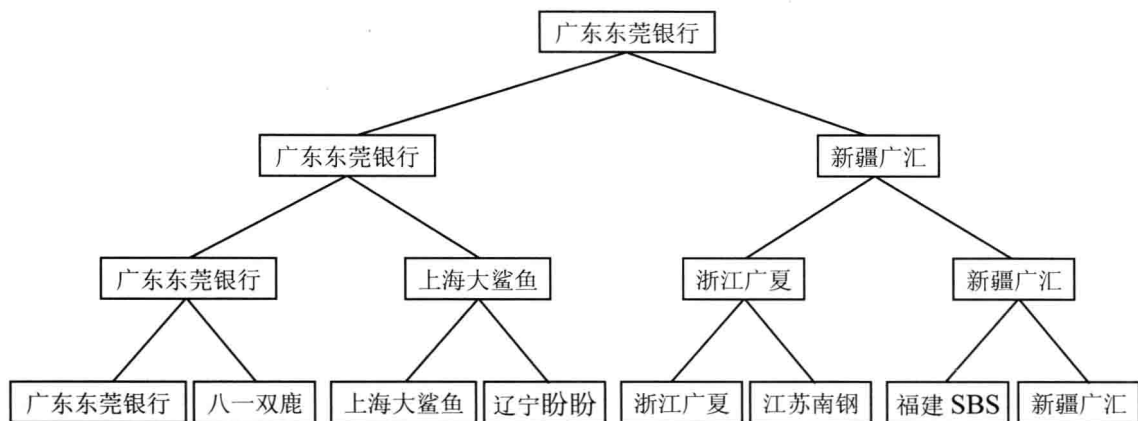


图 1.1 球队对阵形势

【例 1-3】桂林 3 日游。

某旅行社想开辟桂林旅游线路，为降低成本，决定用火车作为交通工具，希望乘车时间越少越好，以便增加游览时间，从而吸引更多游客。

分析：

该问题可用图 1.2 所示的铁路交通图来解决，找出桂林的所有乘车线路中花费时间最少的线路。这类问题的数学模型是图状数据结构。

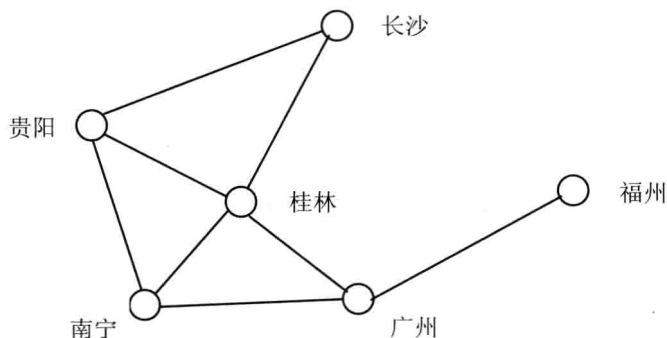


图 1.2 旅游交通图

从上述 3 个例子可以看出，这些发生在人们身边的事都不是数值计算问题，而是非数值计算问题。这些非数值计算问题不能通过列方程、解方程等数学方法来求解，而是用线性表、树、图等数据结构来描述。为了求解这些问题，通常的做法是：先对问题进行抽象，获得表示问题的一个模型；然后围绕该模型设计求解问题的算法；最后用程序实现。获得模型的实质就是分析问题，寻找要操作的对象，以及对象之间的关系。而数据结构正是实际问题中的操作对象，以及这些对象关系的数学抽象。它反映了这些操作对象的内部数据组成，即数据由哪几部分构成、以什么方式构成、呈现什么样的结构、在计算机内如何存储等。因此，数据结构是一门研究非数值计算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的学科。它的主要研究内容如下。

- (1) 数据元素之间固有的逻辑关系——数据逻辑结构。
- (2) 数据元素及关系在计算机内的表示——数据存储结构。

(3) 对数据结构的操作——算法。

任务 1.2 理解相关基本概念和术语

【工作任务】

在了解了数据结构研究的主要内容及范围后，还要理解和掌握数据结构的相关概念和术语。

1. 数据

数据是人们利用文字符号以及其他规定的符号对现实世界的事物及其活动所做的抽象描述，或者说数据是能被计算机识别、存储和加工处理的一切信息。数据一般可分为数值型数据和非数值型数据，如数学中的整数、实数等都是数值型数据，字符、表格、图形、图像和声音等都是非数值型数据。

2. 数据元素

数据元素是数据的基本单位，在计算机程序中通常作为一个整体进行考虑和处理。有时一个数据元素可以由若干个数据项组成，数据项是具有独立含义的最小单位。数据元素有时也称为结点、元素、顶点、记录等。例如，在数据库管理系统中，数据库中的一条记录就是一个数据元素，这条记录中的每一个字段就是构成这个数据元素的数据项。

3. 数据对象

数据对象是性质相同的数据元素的集合，它是数据的一个子集。

4. 数据类型

数据类型是高级程序设计语言中的一个基本概念，是对数据的取值范围、数据的结构以及允许对这些数据进行的操作的一种描述，它规定了程序中操作对象的特性。或者说数据类型是程序设计语言中各变量可取的数据种类。在程序设计语言中，每个变量、常量或表达式都应该属于某种确定的数据类型。

因此，可以把数据类型定义为：数据类型是由若干个值组成的集合以及定义在这个集合上的一组操作的总称，或者说，数据类型是在程序设计语言中已经实现了的数据结构。实际上，数据类型规定了在程序执行期间变量、常量或表达式所有可能的取值范围，以及允许进行的操作。例如，在 C 语言中，如果说一个变量是整数类型的变量，实际上就规定了这个变量的取值范围只能是 $[-\text{maxint}, \text{maxint}]$ 上的整数，其中 maxint 是计算机所允许的最大整数。而且，这个变量在该取值范围内所进行的操作只能是加法、减法、乘法、整除和取余。

数据类型可分为简单类型和结构类型。简单类型中的每个数据都是无法再分割的整体，结构类型是由简单类型按照一定的规则构造而成的，并且结构类型中还可以包括结构类型。例如，在 C 语言中，整数、实数、字符、指针等都是无法再分割的整体，它们所属的类型都是简单类型；数组是一种结构类型，它是由若干个相同类型的数据顺序排列而成的一种数据类型；结构体也是一种结构类型，它是由若干个不同类型的数据顺序排列而成的。

5. 数据结构

数据结构是相互之间存在一种或多种特定关系的数据元素的集合。数据结构包括 3 方面的

内容：数据的逻辑结构、数据的存储结构和数据的操作。

1) 数据的逻辑结构

数据的逻辑结构是指数据元素之间存在的固有的逻辑关系，常简称为数据结构。数据的逻辑结构是从逻辑关系上描述数据的，与数据的存储无关，是独立于计算机的。数据的逻辑结构可以看做是从具体问题抽象出来的数学模型。

依据数据元素之间的关系，可把数据的逻辑结构分为以下 4 种基本类型。

(1) 集合。集合中的数据元素之间除了“同属于一个集合”的关系外，没有其他任何关系。它是数据结构的一种特例，本书不予讨论。

(2) 线性结构。线性结构中的数据元素之间存在“一对一”的关系。若线性结构为非空集合，则除了第一个元素外，其他的每个元素都有唯一的一个直接前驱；除了最后一个元素外，每个元素都有唯一的一个直接后继，如例 1-1 的员工信息表。

(3) 树形结构。树形结构中的数据元素之间存在“一对多”的关系。若树形结构为非空集合，则除第一个数据元素外，其他每个数据元素都只有一个直接前驱，以及零个或多个直接后继，如例 1-2 的球队对阵形势。

(4) 图形结构。图形结构中的数据元素之间存在“多对多”的关系，若图形结构为非空集合，则每个数据元素可以有多个(或零个)直接前驱和多个(或零个)直接后继。

数据结构的 4 种基本类型如图 1.3 所示。有时也可以把数据结构分为两种类型：线性结构和非线性结构。非线性结构包括树形结构和图形结构。

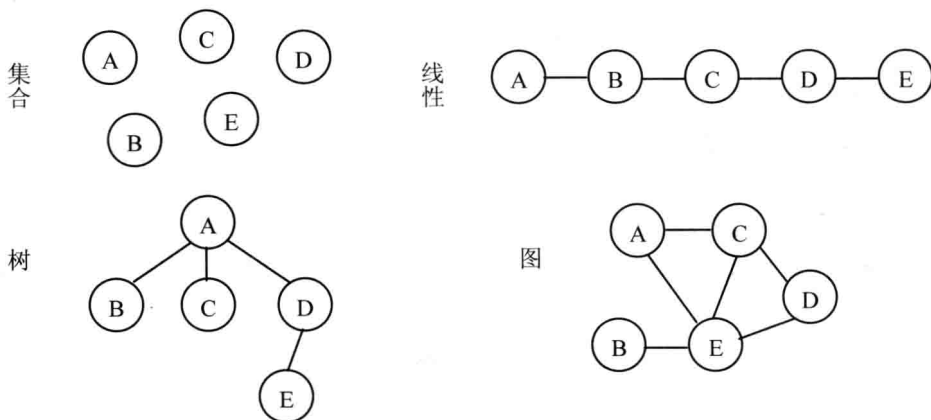


图 1.3 数据逻辑结构

数据结构可以用一个二元组来表示：

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

其中， D 是某个数据对象， R 是该对象中所有数据元素之间的关系的有限集合。

【例 1-4】例 1-1 的数据结构：

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

$D = \{\text{张兵, 杨明, 李婷婷, 王丽丽}\}$,

$R = \{<\text{张兵, 杨明}>, <\text{杨明, 李婷婷}>, <\text{李婷婷, 王丽丽}>\}$ 。

2) 数据的存储结构

数据元素及其关系在计算机内的表示称为数据的存储结构。

要想用计算机处理数据,就必须把数据的逻辑结构映射为数据的存储结构。逻辑结构可以映射为以下 4 种存储结构。

(1) 顺序存储结构。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中,借助元素在存储器中的相对位置来表示数据元素之间的逻辑关系,由此得到的存储结构称为顺序存储结构。计算机的内存单元是一维结构,因此顺序存储结构很方便实现。

(2) 链式存储结构。借助指针表达数据元素之间的逻辑关系。不要求逻辑上相邻的数据元素在物理位置上也相邻,由此得到的存储结构称为链式存储结构。链式存储结构既可以用于实现线性数据结构,也可用于实现非线性数据结构。

(3) 索引存储结构。在存储数据元素的同时,还建立附加的索引表。通过索引表,可以找到存储数据元素的特点。

(4) 散列存储结构。根据散列函数和处理冲突的方法确定数据元素的存储位置。

例如,一个包含数据元素“张兵、杨明、李婷婷、王丽丽”的线性结构,其顺序存储结构和链式存储结构如图 1.4 所示。

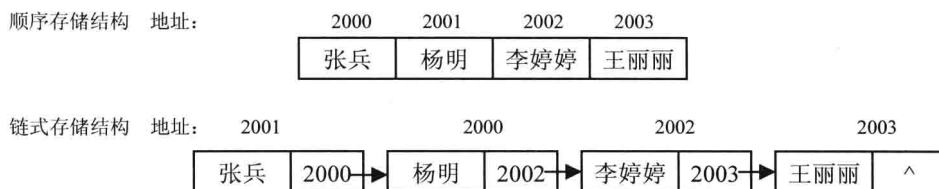


图 1.4 散列存储结构

3) 数据的操作

数据的操作是指在数据的逻辑结构上定义的操作算法,如插入、删除、查找等。

应当注意的是,数据的逻辑结构和数据的物理结构是一个事物的两个方面,而不是两个事物。两者相辅相成,不可分割。一种数据逻辑结构可以映射为多种数据存储结构,具体映射为哪种存储结构,视具体要求而定,主要考虑运算方便及算法的时间和空间效率的要求。

任务 1.3 算 法

【工作任务】

理解算法的概念和进行算法的性能分析。

子任务 1.3.1 理解算法的概念

【课堂任务】理解什么是算法、算法的 5 个特性以及算法描述的不同方式。

1. 算法的概念

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无穷序列,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,解决这个问题的方法有若干种,因此每一种排序方法就是一种算法。

一个算法必须具有以下 5 个特性。

(1) 有穷性。一个算法在执行有限条指令后必须要终止, 且每条指令都要在有限时间内完成。

(2) 确定性。算法中的每条指令必须有确切的含义, 不会产生二义性。

(3) 可行性。一个算法是可行的, 即算法中描述的操作都可以通过执行已经实现的基本运算有限次来实现。

(4) 输入。一个算法有零个或多个输入, 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。

算法和程序有所不同, 程序可以不满足上述的有穷性。例如, Windows 操作系统在用户未操作之前一直处于“等待”的循环中, 直到出现新的用户操作为止。

2. 算法的描述

算法的描述有多种方式。例如, 可以用自然语言、数字语言、图形方式、表格方式或其他约定的符号来描述, 也可以用计算机高级语言来描述, 如 C 语言、Java 语言、C++ 语言等。本书采用 C 语言作为描述算法的工具。

下面通过一个例子来说明算法的不同描述方式。

【例 1-5】已知 n 个整数, 求这 n 个整数中的最大数。

这是一个简单的问题, 下面先用自然语言描述其求解过程。

(1) 将第 1 个数赋值给 max。

(2) 初始化计数变量 i 为 1。

(3) 当 $i < n$ 时, 执行以下代码。

① 比较 $a[i]$ 与 max, 若 $a[i]$ 大于 max, 则将 $a[i]$ 赋值给 max;

② i 自加 1。

(4) 返回 max 的值。

对于上述问题, 现在用 C 语言来描述其求解过程。

```
int Max(int a[], int n)
{
    int i, max;
    max = a[0];
    for(i=1; i<=n; i++)
        if (a[i]>max) max=a[i];
    return
}
```

子任务 1.3.2 算法的性能分析

【课堂任务】了解算法的评价标准, 理解算法的空间复杂度和时间复杂度, 掌握估算算法时间复杂度的方法。

1. 算法的评价标准

对于一个特定的问题, 采用不同的存储结构, 其算法描述一般是不相同的, 即使在同一种存储结构下, 也可以采用不同的求解策略, 从而有许多不同的算法。那么, 对于解决同一问题的不同算法, 选择哪一种算法较为合适, 以及如何对现有的算法进行改进, 从而设计出更好的

算法，这就是算法评价问题。评价一个算法的优劣主要有以下几个标准。

(1) 正确性。一个算法能否正确地执行预先的功能，这是评价一个算法最重要也是最基本的标准。算法的正确性还包括对输入、输出处理的明确的无歧义性的描述。

(2) 可读性。算法主要是为了便于人的阅读理解与交流，其次才是机器执行。即使算法已转变成机器可执行的程序，也需考虑人能较好地阅读理解。可读性好有助于人对算法的理解，这既有助于对算法中隐藏错误的排除，也有助于算法的交流和移植。

(3) 健壮性。算法应具有很强的容错能力，即算法能对非法数据的输入进行检查和处理，不会因非法数据的输入而导致异常中断或死机等现象。

(4) 运行时间。运行时间是指算法在计算机上运行所花费的时间，它等于算法中每条语句执行时间的总和。对于同一个问题如果有多个算法可供选择，应尽可能选择执行时间短的算法。一般说来，执行时间越短，算法的效率越高，性能越好。

(5) 占用空间。占用空间是指算法在计算机存储器上所占用的存储空间，包括存储算法本身所占用的存储空间，算法的输入、输出数据所占用的存储空间和算法运行过程中临时占用的存储空间。算法占用的存储空间是指算法执行过程中所需要的最大存储空间，对于一个问题如果有多个算法可供选择，应尽可能选择存储量需求低的算法。实际上，算法的时间效率和空间效率经常是一对矛盾，相互抵触，有时增加辅助存储空间可以加快算法的运行速度，即用空间换取时间，有时因为内存空间不够，必须压缩辅助存储空间，从而降低了算法的运行速度，即用时间换取空间。

通常把算法在运行过程中临时占用的存储空间的大小叫做算法的空间复杂度。算法的空间复杂度比较容易计算，它主要包括局部变量所占用的存储空间和系统为实现递归所使用的堆栈占用的存储空间。

2. 算法的时间复杂度

一个算法的运行时间是该算法中每条语句执行时间的总和，而每条语句的执行时间是该语句的执行次数(也称为语句频度)与该语句执行一次所需时间的乘积。由于同一条语句在不同的机器上执行所需要的时间是不相同的，也就是说执行一条语句所需的时间与具体的机器有关，所以要想精确地计算出各种语句执行一次所需的时间是比较困难的。实际上，为了评价一个算法的性能，只需计算算法中所有语句执行的总次数即可。

任何一个算法最终都要被分解成一系列基本操作(如赋值、转向、比较、输入、输出等)来具体执行，每一条语句也要分解成具体的基本操作来执行，所以算法的运行时间也可以用算法中所进行的基本操作的总次数来估算。在一个算法中，进行简单操作的次数越少，其运算时间也就相对越少。为了便于比较同一问题的不同算法，也可以用算法中的基本操作重复执行的频度作为算法运行时间的度量标准。

通常把算法中的基本操作重复执行的频度称为算法的时间复杂度。算法中的基本操作一般是指算法最里层循环内的语句，因此，算法中基本操作重复执行的频度 $T(n)$ 是问题规模 n 的某个函数 $f(n)$ ，记作： $T(n)=O(f(n))$ ，其中 O 表示随问题规模 n 的增大，算法执行时间的增长率和 $f(n)$ 的增长率相同，或者说，用 O 符号表示数量级的概念。例如，若 $T(n)=2n^2+5n+7$ ，则 $2n^2+5n+7$ 的数量级与 n^2 的数量级相同，所以 $T(n)=O(n^2)$ 。

如果一个算法没有循环语句，则算法中的基本操作的执行频度与问题规模 n 无关，记作 $O(1)$ ，也称为常数阶。如果一个算法只有一重循环，则算法的基本操作的执行频度随问题规模

n 的增大而呈线性增大关系, 记作 $O(n)$, 也称为线性阶。常用的还有平方阶 $O(n^2)$ 、立方阶 $O(n^3)$ 、对数阶 $O(\log_2 n)$ 、指数阶 $O(2^n)$ 等。这些时间复杂度之间的关系为: $O(1) < O(\log_2 n) < O(n) < O(n \log_2 n) < O(n^2) < O(n^3) < O(n!) < O(n^n)$ 。

下面通过几个例子来说明计算算法时间复杂度的方法。

【例 1-6】 分析以下程序段的时间复杂度。

```
void print( )
{
    int x = 10;
    x++;
    printf("%d", x);
}
```

解: 该算法的基本语句为“ $x++$ ”, 它与问题的规模 n 无关, 因此算法的时间复杂度为 $T(n)=O(1)$ 。

【例 1-7】 分析以下程序段的时间复杂度。

```
void fibonacci(int n)
{
    int fn, f0, f1;
    int i;
    f0=0;
    f1=1;
    for(i=2; i<=n; i++)
    {
        fn=f0+f1;
        printf("%d\t", fn);
        f0=f1;
        f1=fn;
    }
}
```

解: 算法的基本语句是“ $fn=f0+f1$ ”, 其执行次数 $f(n)=n-1 \leq n$, 因此算法的时间复杂度是 $T(n)=O(n)$ 。

【例 1-8】 分析以下程序段的时间复杂度。

```
void sum(int n)
{
    int i, j;
    int tmp, s=0;
    f0=0;
    for(i=1; i<=n; i++)
    {
        tmp=1;
        for(j=1; j<=i; j++) tmp*=j;
        s+=tmp;
        printf("%d\t", s);
    }
}
```

解: 算法的基本语句是“ $tmp*=j$ ”, 其执行次数 $f(n)=1+2+\cdots+n=n(n+1) \leq n^2$, 因此算法的

时间复杂度是 $T(n)=O(n^2)$ 。

【例 1-9】 分析以下程序段的时间复杂度。

```
void print( )
{
    int i;
    for(i=1; i<=n; i*=2)
        printf("%d\t", i);
}
```

解：该算法的基本语句“`printf("%d\t", i)`”，关于执行次数 $f(n)$ ，有 $2^{f(n)} \leq n$ ，即 $f(n) \leq \log_2 n$ ，因此算法的时间复杂度为 $T(n)=O(\log_2 n)$ 。

小 结

本项目主要介绍了数据结构及其相关概念，包括数据、数据元素、数据对象、数据类型、数据结构、逻辑结构、存储结构和算法等基本概念。

根据数据元素之间的不同特性，可以把数据结构分为集合、线性结构、树形结构和图形结构 4 种基本类型。集合中的数据元素之间除了“同属于一个集合”的关系外，没有其他任何关系，它是数据结构的一种特例；线性结构中的数据元素之间存在一个一对一的关系；树形结构中的数据元素之间存在一个一对多的关系；图形结构中的数据元素之间存在多个多对多的关系。有时也可以把数据结构分为两种类型：线性结构和非线性结构。非线性结构包括树形结构和图形结构。

数据的逻辑结构是指数据元素之间的逻辑关系，是用户根据需要而建立起来的；数据的存储结构是指数据元素在计算机的存储器中的存储方式。一般来说，存储结构有 4 种基本类型：顺序存储结构、链式存储结构、Hash 存储结构和索引存储结构。

数据结构一般包括以下 3 方面的内容：数据的逻辑结构、数据的存储结构、对数据元素所进行的操作。

算法是解决特定问题的方法，是由若干条指令组成的有穷序列。一个算法应具有以下 5 个基本特征：有穷性、确定性、可行性、输入和输出；评价一个算法的标准主要有以下 5 个方面：正确性、可读性、健壮性、运行时间、占用的存储空间。

实训：算法时间复杂度分析

1. 实训目的

- (1) 掌握算法时间复杂度的计算方法。
- (2) 了解测试算法运行时间的基本方法。

2. 实训内容

- (1) 分析下列程序的时间复杂度。

程序 1：

```
float sum1(int n)
```

```

{
    float sum=0;
    int i,j;
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            sum+=i*j;
    return sum;
}

```

程序 2:

```

float sum2(int n)
{
    float sum=0;
    int i=0;
    while(i<n)
    {
        sum+=i;
        i+=2;
    }
    return sum;
}

```

(2) 使用 C 语言的标准库函数 `ftime`, 计算以上程序的运行时间, 并与分析得出的时间复杂度进行比较。

习 题

一、选择题

1. 数据结构是一门研究非数值计算程序设计问题中计算机的()以及它们之间的()和操作的学科。

- ①A. 操作对象 B. 计算方法 C. 逻辑存储 D. 数据映像
②A. 结构 B. 关系 C. 运算 D. 算法

2. 数据结构被形式定义为(D, R), 其中 D 是()的有限集合, R 是 D 上的()的有限集合。

- ①A. 算法 B. 数据元素 C. 数据操作 D. 逻辑结构
②A. 操作 B. 映像 C. 存储 D. 关系

3. 在数据结构中, 从逻辑上可以把数据结构分成()。

- A. 动态结构和静态结构 B. 紧凑结构和非紧凑结构
C. 线性结构和非线性结构 D. 内部结构和外部结构

4. 线性表的顺序存储结构是一种()的存储结构, 线性表的链式存储结构是一种()的存储结构。

- A. 随机存取 B. 顺序存取 C. 索引存取 D. 散列存取

5. 进行算法分析的目的是(), 算法分析的两个主要方面是()。

- ①A. 找出数据结构的合理性 B. 研究算法中的输入和输出的关系
C. 分析算法的效率以求改进 D. 分析算法的易懂性和文档性