

● 高等学校21世纪计算机教材

# 数据结构

## (C语言版)

周云静 编著

.12

冶金工业出版社

高等学校 21 世纪计算机教材

# 数 据 结 构

(C 语言版)

周云静 编著

北 京

冶 金 工 业 出 版 社

2003

## 内 容 简 介

本书全面介绍了各种数据结构及其相关的算法,详细阐述了每种数据结构的基本概念。在算法实现上讲求示范性、通俗性;对于重要的算法,给出了C语言实现的源程序。大部分章节都有应用实例,供学习相应内容时参考。

本书最大的特点是通俗易懂,讲求前后内容的连贯性。每章后面附有大量精选的习题,并在最后给出了参考答案,以便读者自我检验。

本书可作为计算机专业、信息专业或其他相关专业的本、专科教材,也是广大参加自学考试的人员和软件开发工作者的参考资料。本书既可作为“数据结构”课程的教材,也可作为其他程序设计类课程的辅导教材。

### 图书在版编目(CIP)数据

数据结构: C语言版 / 周云静编著. —北京: 冶金工业出版社, 2003.5

ISBN 7-5024-3258-2

I. 数... II. 周... III. ①数据结构 ②C语言—程序设计 IV. ①TP311.12 ②TP312

中国版本图书馆 CIP 数据核字 (2003) 第 026709 号

出版人 曹胜利 (北京沙滩嵩祝院北巷 39 号, 邮编 100009)

责任编辑 程志宏

中山市新华印刷厂有限公司印刷; 冶金工业出版社发行; 各地新华书店经销

2003 年 6 月第 1 版, 2003 年 6 月第 1 次印刷

787mm×1092mm 1/16; 21.25 印张; 518 千字; 332 页; 1-5000 册

28.00 元

冶金工业出版社发行部 电话: (010) 64044283 传真: (010) 64027893

冶金书店 地址: 北京东四西大街 46 号 (100711) 电话: (010) 65289081

(本社图书如有印装质量问题, 本社发行部负责退换)

# 前 言

## 一、数据结构理论的发展

数据结构理论发展至今,已经成为一门比较成熟的课程,数据结构是计算机系统软件和应用软件开发者的必修课程。数据结构理论的应用范围已经渗透到编译系统、操作系统、数据库、人工智能、信息科学、企业管理、系统工程、计算机辅助设计及其他信息管理的应用中。

## 二、本书特点

本书采用 C 语言描述。在介绍时,着重强调理论与实践的结合,避免空洞的概念讲解,注重可读性、实用性、易懂性。在教学内容的组织上沿着“能做什么——如何做——做得如何”的思路进行。在介绍各种数据结构理论时,首先介绍它的应用背景,让读者明确这些知识在实际工作中的应用价值;然后,在介绍算法时,重点突出实现的过程与方法,不仅详细介绍了算法思想,而且给出重点算法的完整 C 语言源代码,突出示范性,帮助读者进一步理解如何在程序设计中实现算法思想,把理论与实践紧密结合起来,深刻理解数据结构理论的内涵;最后体现各知识点在实际应用中的效果,在重要知识点的后面附有精心设计的应用实例,把相关知识点的算法都融入到实例中,通过实际的例子,调动学生模仿练习的学习积极性,从而达到提高程序设计的能力。

本书注重前后内容的相互联系,引导读者循序渐进的掌握数据结构课程的内容。为了让读者巩固所学知识,每章后面都有大量精心设计的习题,读者可以对照参考答案进行练习,以此巩固自己所学的知识。

## 三、本书结构

本书共分 10 章,其结构如下:

第 1 章:绪论。本章内容是后续章节的一个铺垫,主要介绍了数据结构的基本概念、数据结构的重要性、算法评价、算法分析等内容。其中算法分析中的时间复杂度分析是贯穿全书的一根线索。

第 2 章:C 语言基础。本章介绍了 C 语言的基本知识,主要包括 C 语言的数据类型、C 语言的常用运算符、结构化程序设计、算法模块化思想、培养良好的程序设计风格等内容。

第 3 章:基本线性表。基本线性表是最常用、最简单的一种数据结构类型。本章主要介绍了基本线性表的应用背景、基本线性表逻辑结构、基本线性表的顺序表示及实现、基本线性表的链式表示及实现,最后通过实例介绍了基本线性表的应用。

第 4 章:特殊线性表。特殊线性表是与基本线性表有所区别,但操作上又有相似之处的另一类典型数据结构。本章主要介绍了特殊线性表应用背景、队列、堆栈、字符串,最后通过实例介绍了特殊线性表的应用。

第 5 章:树与二叉树。树与二叉树是一种有明显层次结构的、应用广泛的典型数据结构。本章主要介绍了树的应用背景,树,二叉树,二叉树与树、森林的转换,二叉排序树,哈夫曼树,最后通过实例介绍了树与二叉树的应用。

第6章：查找。查找既是数据结构中的一种重要的数据处理技术，也是在基本线性表上的具体应用。本章主要介绍了查找技术应用背景、基本概念和术语、线性表查找、树型查找、Hash表查找，最后通过实例介绍了查找的应用。

第7章：排序。排序是数据结构理论中的另一个数据处理的重点，通常也是在基本线性表上展开。本章主要介绍了排序的应用背景、基本概念、插入排序、选择排序、交换排序、归并排序、基数排序、内部排序、外部排序，最后通过实例介绍了排序的应用。

第8章：数组、矩阵和广义表。数组、矩阵和广义表是基本线性表内容的延伸和扩展。同时也是第9章“图”的基础。因为“图”存储和运算都是以矩阵为表示形式的。本章主要介绍了数组的应用背景、多维数组、特殊矩阵的压缩存储、广义表。

第9章：图。图是又一类典型的数据结构，是数据结构理论的难点，同时也是数据理论的精华部分。运用图的知识可以解决许多复杂的实际问题。本章主要介绍了图的应用背景、图的基本概念和术语、图的存储结构、图的遍历、图的连通性、最小生成树、最短路径、有向无环图，最后通过实例介绍了图的应用。

第10章：文件。本章主要介绍了一些不同类型文件的组织形式及其特点，包括文件的基本概念、顺序文件、索引文件、散列文件、多关键字文件等内容。

在本书的最后还给出了每章习题的参考答案，以供读者参考。

#### 四、本书学习指导

建议学习本书为70学时，其中20学时为上机实践。根据不同的情况，教学内容和学时数可以作适当调整。对于C语言基础较好的学生，可以跳过第2章，如果没有C语言基础，则可详细学习第2章，并适当增加总教学学时数。

#### 五、本书适用对象

本书可作为计算机专业、信息专业或其他相关专业的本、专科教材，也是广大参加自学考试的人员和软件开发工作者的参考资料。本书既可作为“数据结构”课程的教材，也可作为其他程序设计类课程的辅导教材。

本书由周云静编写，方平审校。在编写过程中，得到了家人和同事的大力支持和鼓励，在此表示衷心的感谢。如果读者在阅读本书的过程中遇到疑难问题或觉得不妥之处，可到相关网站的论坛进行讨论，网址：<http://www.cnbook.net>。

由于编写时间仓促，水平有限，书中疏漏之处在所难免，敬请读者批评指正。

编 者

2003年4月

# 目 录

|                               |    |                            |    |
|-------------------------------|----|----------------------------|----|
| <b>第1章 绪论</b> .....           | 1  | 2.4 算法模块化思想.....           | 47 |
| 1.1 数据结构的基本概念.....            | 1  | 2.4.1 模块化思想在算法设计中的重要性..... | 47 |
| 1.1.1 数据结构的含义.....            | 1  | 2.4.2 使用函数模块化.....         | 47 |
| 1.1.2 数据结构的基本术语.....          | 3  | 2.5 培养良好的程序设计风格.....       | 50 |
| 1.2 数据结构的重要性.....             | 6  | 2.5.1 明确需求分析.....          | 51 |
| 1.2.1 数据结构理论的发展历史.....        | 6  | 2.5.2 加强程序可读性.....         | 51 |
| 1.2.2 数据结构在计算机课程体系中的重要地位..... | 6  | 2.5.3 使用有意义的变量名、函数名.....   | 52 |
| 1.3 算法评价.....                 | 7  | 小结.....                    | 52 |
| 1.3.1 算法的定义及表示.....           | 7  | 综合练习二.....                 | 53 |
| 1.3.2 算法的特征及评价方法.....         | 9  | 一、选择题.....                 | 53 |
| 1.4 算法分析.....                 | 10 | 二、填空题.....                 | 56 |
| 1.4.1 算法的时间复杂度分析.....         | 10 | 三、算法及程序设计.....             | 57 |
| 1.4.2 算法的空间复杂度分析.....         | 14 | <b>第3章 基本线性表</b> .....     | 58 |
| 小结.....                       | 15 | 3.1 应用背景.....              | 58 |
| 综合练习一.....                    | 15 | 3.2 基本线性表的逻辑结构.....        | 58 |
| 一、选择题.....                    | 15 | 3.2.1 基本线性表的定义.....        | 58 |
| 二、填空题.....                    | 16 | 3.2.2 基本线性表的运算.....        | 59 |
| 三、阅读理解题.....                  | 17 | 3.3 基本线性表的顺序表示及实现.....     | 60 |
| <b>第2章 C语言基础</b> .....        | 19 | 3.3.1 基本线性表的顺序存储.....      | 60 |
| 2.1 C语言的数据类型.....             | 19 | 3.3.2 基本线性表顺序存储的运算.....    | 62 |
| 2.1.1 基本数据类型的使用.....          | 19 | 3.4 基本线性表的链式表示及实现.....     | 68 |
| 2.1.2 常用构造数据类型的使用.....        | 23 | 3.4.1 基本线性表的链式存储.....      | 68 |
| 2.1.3 指针数据类型的使用.....          | 27 | 3.4.2 基本线性表的单链表操作.....     | 69 |
| 2.2 C语言的常用运算符.....            | 33 | 3.4.3 基本线性表的双链表操作.....     | 75 |
| 2.2.1 算术运算符的使用.....           | 33 | 3.4.4 基本线性表的循环链表操作.....    | 77 |
| 2.2.2 逻辑运算符与关系运算符的使用..        | 34 | 3.5 应用实例.....              | 77 |
| 2.2.3 条件运算符与逗号运算符的使用..        | 35 | 小结.....                    | 83 |
| 2.2.4 特殊运算符的使用.....           | 36 | 综合练习三.....                 | 83 |
| 2.2.5 Turbo C 运算符的优先级.....    | 37 | 一、选择题.....                 | 83 |
| 2.3 结构化程序设计.....              | 37 | 二、填空题.....                 | 84 |
| 2.3.1 结构化程序设计思想.....          | 37 | 三、算法及程序设计.....             | 85 |
| 2.3.2 顺序结构程序设计.....           | 38 | <b>第4章 特殊线性表</b> .....     | 86 |
| 2.3.3 选择结构程序设计.....           | 38 | 4.1 应用背景.....              | 86 |
| 2.3.4 循环结构程序设计.....           | 41 | 4.2 队列.....                | 87 |

|                          |            |                        |            |
|--------------------------|------------|------------------------|------------|
| 4.2.1 队列的含义.....         | 87         | 关系 .....               | 131        |
| 4.2.2 队列的顺序存储及运算 .....   | 88         | 5.3.6 二叉树的基本操作 .....   | 133        |
| 4.2.3 队列的链式存储及运算 .....   | 91         | 5.3.7 二叉树的线索化 .....    | 135        |
| 4.2.4 循环队列的表示及实现 .....   | 92         | 5.4 二叉树与树、森林的转换 .....  | 137        |
| 4.3 堆栈 .....             | 96         | 5.4.1 二叉树、树与森林 .....   | 137        |
| 4.3.1 堆栈的含义 .....        | 96         | 5.4.2 树、森林的遍历 .....    | 138        |
| 4.3.2 堆栈的顺序存储及运算 .....   | 97         | 5.5 二叉排序树 .....        | 139        |
| 4.3.3 堆栈的链式存储结构及运算 ..... | 99         | 5.5.1 二叉排序树的定义 .....   | 139        |
| 4.3.4 双栈的表示及实现 .....     | 100        | 5.5.2 二叉排序树的运算 .....   | 140        |
| 4.4 堆栈与队列的比较 .....       | 101        | 5.6 哈夫曼树 .....         | 144        |
| 4.4.1 堆栈和队列的相同点 .....    | 102        | 5.6.1 哈夫曼树的基本概念 .....  | 144        |
| 4.4.2 堆栈和队列的不同点 .....    | 102        | 5.6.2 哈夫曼树的构造 .....    | 145        |
| 4.5 字符串 .....            | 102        | 5.6.3 哈夫曼树的应用 .....    | 146        |
| 4.5.1 字符串的基本概念 .....     | 103        | 5.7 应用实例 .....         | 149        |
| 4.5.2 字符串的顺序存储及运算 .....  | 103        | 小结 .....               | 152        |
| 4.5.3 字符串的链式存储及运算 .....  | 105        | 综合练习五 .....            | 152        |
| 4.5.4 字符串的混合存储及表示 .....  | 107        | 一、选择题 .....            | 152        |
| 4.6 应用实例 .....           | 108        | 二、填空题 .....            | 154        |
| 4.6.1 队列的应用 .....        | 108        | 三、算法及程序设计 .....        | 154        |
| 4.6.2 堆栈的应用 .....        | 109        | <b>第6章 查找 .....</b>    | <b>155</b> |
| 4.6.3 字符串的应用 .....       | 109        | 6.1 应用背景 .....         | 155        |
| 小结 .....                 | 111        | 6.2 基本概念和术语 .....      | 155        |
| 综合练习四 .....              | 112        | 6.3 线性表查找 .....        | 156        |
| 一、选择题 .....              | 112        | 6.3.1 顺序查找 .....       | 156        |
| 二、填空题 .....              | 113        | 6.3.2 折半查找 .....       | 159        |
| 三、算法及程序设计 .....          | 113        | 6.3.3 分块查找 .....       | 161        |
| <b>第5章 树与二叉树 .....</b>   | <b>114</b> | 6.4 树型查找 .....         | 162        |
| 5.1 应用背景 .....           | 114        | 6.4.1 二叉排序树的查找算法 ..... | 162        |
| 5.2 树 .....              | 115        | 6.4.2 平衡二叉树的查找算法 ..... | 163        |
| 5.2.1 树的基本概念与术语 .....    | 115        | 6.4.3 B-树查找算法 .....    | 169        |
| 5.2.2 树的存储与表示 .....      | 117        | 6.4.4 B+树查找算法 .....    | 171        |
| 5.2.3 树的性质 .....         | 120        | 6.4.5 B*树查找算法 .....    | 172        |
| 5.3 二叉树 .....            | 121        | 6.5 Hash 表查找 .....     | 172        |
| 5.3.1 二叉树的基本概念 .....     | 121        | 6.5.1 Hash 表的内涵 .....  | 172        |
| 5.3.2 二叉树的性质 .....       | 121        | 6.5.2 Hash 函数的构造 ..... | 173        |
| 5.3.3 二叉树的存储与建立 .....    | 122        | 6.5.3 关键字冲突处理方法 .....  | 174        |
| 5.3.4 二叉树的遍历 .....       | 125        | 6.6 应用实例 .....         | 176        |
| 5.3.5 二叉树遍历与二叉树构造的       |            | 小结 .....               | 178        |

|                           |            |                                |            |
|---------------------------|------------|--------------------------------|------------|
| 综合练习六.....                | 178        | 8.2 多维数组.....                  | 214        |
| 一、选择题.....                | 178        | 8.2.1 多维数组的逻辑结构.....           | 214        |
| 二、填空题.....                | 180        | 8.2.2 多维数组的顺序存储.....           | 215        |
| 三、算法及程序设计.....            | 180        | 8.3 特殊矩阵的压缩存储.....             | 217        |
| <b>第7章 排序.....</b>        | <b>181</b> | 8.3.1 对称矩阵.....                | 217        |
| 7.1 应用背景.....             | 181        | 8.3.2 三角矩阵.....                | 218        |
| 7.2 基本概念.....             | 181        | 8.3.3 带状矩阵.....                | 219        |
| 7.3 插入排序.....             | 183        | 8.3.4 稀疏矩阵.....                | 220        |
| 7.3.1 直接插入排序.....         | 183        | 8.4 广义表.....                   | 230        |
| 7.3.2 折半插入排序.....         | 184        | 8.4.1 广义表的定义和基本运算.....         | 230        |
| 7.3.3 希尔排序.....           | 185        | 8.4.2 广义表的存储.....              | 233        |
| 7.4 选择排序.....             | 187        | 8.4.3 广义表基本操作的实现.....          | 235        |
| 7.4.1 简单选择排序.....         | 187        | 小结.....                        | 237        |
| 7.4.2 树型选择排序.....         | 188        | 综合练习八.....                     | 238        |
| 7.4.3 堆排序.....            | 189        | 一、选择题.....                     | 238        |
| 7.5 交换排序.....             | 191        | 二、填空题.....                     | 239        |
| 7.5.1 冒泡排序.....           | 191        | 三、算法及程序设计.....                 | 239        |
| 7.5.2 快速排序.....           | 192        | <b>第9章 图.....</b>              | <b>240</b> |
| 7.6 归并排序.....             | 195        | 9.1 应用背景.....                  | 240        |
| 7.7 基数排序.....             | 197        | 9.2 图的基本概念和术语.....             | 240        |
| 7.7.1 多关键字排序.....         | 197        | 9.3 图的存储结构.....                | 243        |
| 7.7.2 链式基数排序.....         | 198        | 9.3.1 邻接矩阵表示.....              | 243        |
| 7.8 内部排序.....             | 201        | 9.3.2 邻接表表示.....               | 245        |
| 7.8.1 内部排序方法的共同点.....     | 201        | 9.3.3 十字链表表示.....              | 247        |
| 7.8.2 内部排序方法的不同点.....     | 201        | 9.3.4 邻接多重表表示.....             | 249        |
| 7.8.3 排序方法的选择.....        | 202        | 9.4 图的遍历.....                  | 250        |
| 7.9 外部排序.....             | 202        | 9.4.1 深度优先搜索.....              | 250        |
| 7.9.1 外部排序的方法.....        | 202        | 9.4.2 广度优先搜索.....              | 252        |
| 7.9.2 多路平衡归并的实现.....      | 204        | 9.4.3 广度和深度优先搜索的实现.....        | 253        |
| 7.10 应用实例.....            | 206        | 9.5 图的连通性.....                 | 256        |
| 小结.....                   | 210        | 9.5.1 无向图的连通性.....             | 256        |
| 综合练习七.....                | 211        | 9.5.2 有向图的连通性.....             | 257        |
| 一、选择题.....                | 211        | 9.5.3 生成树和生成森林.....            | 257        |
| 二、填空题.....                | 213        | 9.5.4 连通性的应用.....              | 259        |
| 三、算法及程序设计.....            | 213        | 9.6 最小生成树.....                 | 260        |
| <b>第8章 数组、矩阵和广义表.....</b> | <b>214</b> | 9.6.1 最小生成树的概念.....            | 260        |
| 8.1 应用背景.....             | 214        | 9.6.2 构造最小生成树的 Prim 算法.....    | 261        |
|                           |            | 9.6.3 构造最小生成树的 Kruskal 算法..... | 263        |

|                        |            |                         |            |
|------------------------|------------|-------------------------|------------|
| 9.7 最短路径.....          | 265        | 10.3.3 索引文件的操作.....     | 294        |
| 9.7.1 Dijkstra 算法..... | 265        | 10.3.4 利用查找表建立多级索引..... | 295        |
| 9.7.2 Floyd 算法.....    | 267        | 10.4 散列文件.....          | 295        |
| 9.8 有向无环图.....         | 269        | 10.5 多关键字文件.....        | 297        |
| 9.8.1 有向无环图的概念及应用..... | 269        | 10.5.1 多关键字文件概述.....    | 297        |
| 9.8.2 AOV 网与拓扑排序.....  | 270        | 10.5.2 多重表文件.....       | 297        |
| 9.8.3 AOE 网与关键路径.....  | 274        | 10.5.3 倒排文件.....        | 298        |
| 9.9 应用实例.....          | 278        | 小结.....                 | 300        |
| 小结.....                | 287        | 综合练习十.....              | 300        |
| 综合练习九.....             | 288        | 一、选择题.....              | 300        |
| 一、选择题.....             | 288        | 二、填空题.....              | 301        |
| 二、填空题.....             | 289        | 三、算法及程序设计.....          | 302        |
| 三、算法及程序设计.....         | 289        | <b>参考答案.....</b>        | <b>303</b> |
| <b>第 10 章 文件.....</b>  | <b>290</b> | 第 1 章.....              | 303        |
| 10.1 文件的基本概念.....      | 290        | 第 2 章.....              | 304        |
| 10.1.1 文件的概念.....      | 290        | 第 3 章.....              | 307        |
| 10.1.2 文件的逻辑结构及操作..... | 290        | 第 4 章.....              | 310        |
| 10.1.3 文件的存储结构.....    | 291        | 第 5 章.....              | 313        |
| 10.2 顺序文件.....         | 292        | 第 6 章.....              | 316        |
| 10.2.1 顺序文件的定义及分类..... | 292        | 第 7 章.....              | 318        |
| 10.2.2 顺序文件的操作.....    | 292        | 第 8 章.....              | 321        |
| 10.3 索引文件.....         | 293        | 第 9 章.....              | 324        |
| 10.3.1 索引文件的定义及构成..... | 293        | 第 10 章.....             | 328        |
| 10.3.2 索引文件的存储.....    | 294        | <b>参考文献.....</b>        | <b>332</b> |

# 第1章 绪 论

## 本章要点

- 数据结构的基本概念
- 数据结构的重要性
- 算法评价
- 算法分析

随着计算机（特别是微型计算机）技术和应用的飞速发展，其应用领域已经延伸到人们生活的各个方面。在教育、商业、金融、工矿企业、科学研究、军事、邮电、通信，甚至家庭生活都离不开计算机了。借助计算机，许多行业和产业都成倍的提高了生产力和生产效率。

计算机所处理的数据对象也从以前的数值计算扩展到非数值计算，如银行业务处理、行政管理、计算机辅助设计、图形图像处理、人工智能等。面对不同的数据处理对象，不同的需求，数据的组织形式、存储及运算必须有不同的方法，才能进行有效的处理。因此努力研究数据的内在结构，才能寻找到某些规律性的方法。对于数据结构的深入研究，在应用领域中有着广泛的需求。

## 1.1 数据结构的基本概念

### 1.1.1 数据结构的含义

数据结构和算法是程序设计最重要的两个内容。

那么什么是数据结构呢？简单地说，数据结构是数据的组织、存储和运算的总和。它是信息的一种组织方式，是以数据按某种组织关系联系起来的一批数据，其目的是为了提高算法的效率，然后用一定的存储方式存储到计算机中，并且它通常与一组算法的集合相对应，通过这组算法集合可以对数据结构中的数据进行某种操作。

在计算机处理的大量数据中，它们都是相互关联，彼此联系的。

下面以航班查询系统为例进行说明，顾客可以到机票销售网点去查询机票情况，也可以上互联网进行查询。查询时，如果找当天某个特定的航班，可以输入被查询航班号，如图 1-1 所示。

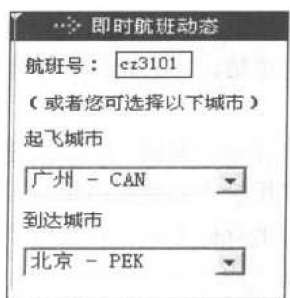


图 1-1 单个条件航班号查询

在“即时航班动态”查询中输入航班号 CZ3101, 当天航班 CZ3101 的相应信息马上就在电脑屏幕上显示出来, 如图 1-2 所示。

广州-北京 2002-11-7 9:27:45 的航班动态信息

| 航班号    | 机型    | 始发地  | 目的地  | 预计起飞时间  | 预计到达时间   | 实际起飞时间  | 实际到达时间 | 数据刷新时间  | 备注 |
|--------|-------|------|------|---------|----------|---------|--------|---------|----|
| CZ3101 | 波音777 | 白云机场 | 首都机场 | 8:20:00 | 11:05:00 | 8:15:00 |        | 9:27:39 |    |

图 1-2 单个条件查询结果

如果顾客查询的不是当天的航班信息, 而是准备预定以后的某个航班座位, 并且不知道航班号, 只确定航班的起飞地点和到达地点, 同样也可以找到所需的航班信息。如图 1-3 所示, 在查询系统中输入起始城市为“广州”, 到达城市为“北京”, 起飞日期为“2002-11-7”, 那么, 符合查询条件的所有航班信息都会在屏幕上显示出来。如图 1-4 所示是查询的结果。

航班号

起始城市

起飞日期

到达城市

返回日期

航班号:

(或者您可选择以下城市)

起飞城市

到达城市

查询日期

图 1-3 复合条件航班查询

航班时刻表 航段一 广州 至 北京 2002年11月7日(星期四) 航班的执行情况

| 航班号    | 起始时间 | 到达时间 | 星期      | 机型    | 经停 | 航班执行起始日期   | 航班执行终止日期  |
|--------|------|------|---------|-------|----|------------|-----------|
| CZ3101 | 0820 | 1105 | 1234567 | 波音777 | 0  | 2001-11-5  | 2003-1-23 |
| CZ3105 | 0820 | 1210 | 1234567 | 波音757 | 0  | 2002-11-1  | 2003-3-29 |
| CZ3103 | 1120 | 1405 | 1234567 | 波音777 | 0  | 2002-10-27 | 2003-1-23 |
| CZ3111 | 1330 | 1610 | 1234567 | 波音757 | 0  | 2002-10-28 | 2003-3-29 |
| CZ3109 | 1520 | 1805 | 1234567 | 波音777 | 0  | 2002-10-27 | 2003-1-23 |

图 1-4 复合条件航班查询结果

从以上两种不同查询方法的结果可以看出, 每一个航班的基本情况都由航班号、机型、起飞时间、到达时间等多项内容。这些类型的数据本来互不相同, 但组织在一起, 就构成了一个记录。每一个航班的基本情况都有一个记录相对应。如果要查找某个航班的机型、起飞时间、到达时间等就不必一一去查询了, 只要在计算机内查询航班号, 该航班的其他基本信息就会一起显示出来, 航班信息一目了然。要查找某天的航班情况, 只要输入日期到计算机内, 相应日期的航班情况也就清清楚楚, 大大提高了操作效率。这就是数据结构对数据合理组织、合理设计运算方法的作用。

当然每个航班信息所包含的内容非常繁琐, 要想快速查询到相应信息, 首先要弄清航班信息所包含数据的逻辑结构, 才能用恰当的数据组织方式表示出来。再根据航班信息动态变化的实际, 在计算机内部设计合理的存储方式。仅这两个方面还不够, 还要有相应的运算方法, 也就是要设计一个合理的数据结构。

在现实生活中, 诸如航班查询这一类的例子比比皆是。如销售查询、图书查询, 储蓄查

询、流动人口查询、学历查询等等。对于这一类的问题，数据元素之间的逻辑关系是线性的，可以用称为线性表的结构来组织数据。线性表是数据结构中最常用，最简单的一类典型数据结构。

有时还会碰到一些更为复杂的问题，如编译系统中的“指令代码文法分析和翻译”，下棋游戏中的“对策处理”，电报电文的代码编译等，这一类问题都是用“树型”的数学模型来研究它的数据组织及运算。这是数据结构理论中所研究的一类较复杂的典型数据结构。

对于工程工期合理规划、最小工程成本、交通灯管理、动态规划这一类的应用问题，解决的方法通常采用数据结构中“图”这一类模型来研究它的数据组织、存储及运算方法。

由此看来，数据结构理论研究的内容不再是简单的数学方程，而是非数值计算问题的数学模型，诸如表、树和图之类的数据结构，因此，简单说来，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操作对象以及它们之间的关系和操作的学科。

数据结构是在整个计算机科学与技术领域上广泛使用的术语。它用来反映一个数据的内部构成，即一个数据由哪些成分数据构成，以什么方式构成，呈什么结构。数据结构有逻辑上的数据结构和物理上的数据结构之分。逻辑上的数据结构反映成分数据之间的逻辑关系，而物理上的数据结构反映成分数据在计算机内部的存储安排。数据结构是数据存在的形式。

数据结构作为一门学科主要研究数据的各种逻辑结构和存储结构，以及对数据的各种操作。因此，主要有三个方面的内容：数据的逻辑结构；数据的物理存储结构；对数据的操作（或算法）。通常，算法的设计取决于数据的逻辑结构，算法的实现取决于数据的物理存储结构。

### 1.1.2 数据结构的基本术语

数据结构的基本术语是学习数据结构的基础。因此首先有必要弄清楚相关的一些术语。

#### 1. 数据 (Data)

数据即信息的载体，是对客观事物的符号表示，指能输入到计算机中并被计算机程序处理的符号的总称。如整数、实数、字符、文字、声音、图形、图像等都是数据。

#### 2. 数据元素 (Data Element)

数据元素是数据的基本单位，它在计算机处理和程序设计中通常作为独立个体考虑的对象。数据元素一般由一个或多个数据项组成。一个数据元素包含多个数据项时，常称为记录、结点等。数据项也称为域、字段、属性、表目、顶点。

#### 3. 数据对象 (Data Object)

数据对象是具有相同特征的数据元素的集合，是数据的一个子集，如行政管理中员工的全部资料、销售系统中与产品相关联的全部资料。

#### 4. 数据结构 (Data Structure)

数据结构简称 DS，是数据元素的组织形式，或数据元素相互之间存在一种或多种特定关系的集合。任何数据都不是彼此孤立的，通常把相关联的数据按照一定的逻辑关系组织起来，按照计算机语言的语法、语义的规则定义相应的存储结构或形式，并且为这些数据指定一组运算操作，这样就形成了一个数据结构。数据结构可用二元组的形式表示为：

$$B = (D, S)$$

其中：D 是数据元素的有限集，S 是 D 上关系的有限集。数据结构通常有四类基本形式：

集合结构、线性结构、树型结构、图形结构或网状结构。

在集合结构中,各个数据元素之间不存在关系(如图 1-5 所示),这是数据结构的一种特例,本书不予讨论。

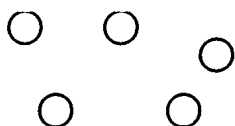


图 1-5 集合结构

在线性结构中,除首元结点外,每个结点都有一个惟一的前驱,除尾结点外,每个结点都有惟一的后继。数据元素之间存在着一对一的关系,如图 1-6 所示。



图 1-6 线性结构

在树型结构中,除根结点外,每个结点都有一个惟一的前驱,但可以有多个后继。数据元素之间存在着一对多的关系,如图 1-7 所示。

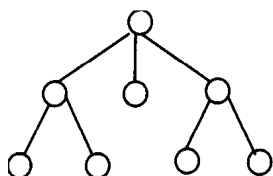


图 1-7 树型结构

在图形结构中,每个结点可以有多个前驱和任意多个后继。数据元素之间存在着多对多的关系,如图 1-8 所示。通常把树型和图形结构称为非线性结构。

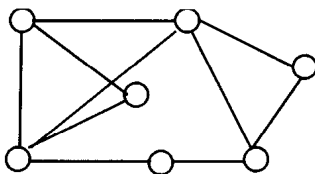


图 1-8 图形结构

## 5. 数据的逻辑结构 (Logical Structure)

是指数据结构中数据元素之间的逻辑关系。它是从具体问题中抽象出来的数学模型,是独立于计算机存储器的(与具体的计算机无关)。如果数据结构 DS 用一个二元组表示为:

$$DS = (D, S)$$

这里, S 是定义在 D (或其他集合) 上的关系的集合,  $S = \{ R \mid R: D \times D \times \dots \}$ , 那么 S 就是数据的逻辑结构。数据的逻辑结构可分为四种基本类型: 集合结构、线性结构、树型结构和图形结构。表和树是最常用的两种高效数据结构, 许多高效的算法可以用这两种数据结构来设计实现。表是线性结构(全序关系), 树(偏序或层次关系)和图(局部有序“weak/local orders”)是非线性结构。

## 6. 数据的存储结构 (Physical Structure)

数据的存储结构是数据的逻辑结构在计算机内存中的存储方式, 又称物理结构。数据存

储结构的实现要用计算机编程语言来实现，因而是依赖于具体的计算机语言。数据存储结构有顺序和链式两种不同的方式。顺序存储结构的特点是要借助数据元素在存储器中的相对位置来体现数据元素相互间的逻辑关系。而链式存储结构则通过表示数据元素存储地址的指针来表示数据元素之间的逻辑关系。顺序存储结构通常用高级编程语言中的“一维数组”来描述或实现。而链式存储结构则通常用链表来实现。例如集合  $A=\{2, 4, 6, 8\}$  的顺序存储结构如图 1-9 所示。链式存储结构如图 1-10 所示。在实际应用中，解决不同的问题时，设计不同的算法要依据数据的逻辑结构，但算法如何实现时又要取决于存储结构。

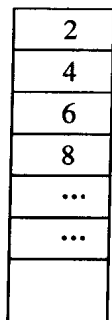


图 1-9 顺序存储结构

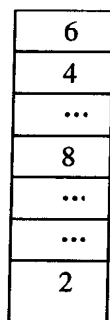


图 1-10 链式存储结构

在顺序存储结构的基础上，又可延伸变化出另外两种存储结构，即索引存储和散列存储。

索引存储就是在数据文件的基础上增加了一个索引表文件。通过索引表建立索引，可以把一个顺序表分成几个顺序子表，其目的是在查询时提高查找效率，避免盲目查找。如一个家族用索引存储可表示为如图 1-11 所示的形式。

散列存储就是通过数据元素与存储地址之间建立起某种映射关系，使每个数据元素与每一个存储地址之间尽量达到一一对应的目的。这样，查找时同样可大大提高效率。如集合  $A=\{2, 5, 7, 9, 11\}$ ，通过  $f(y)=y \bmod 17$  得到的散列存储结构如图 1-12 所示。

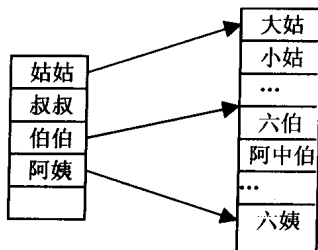


图 1-11 索引存储

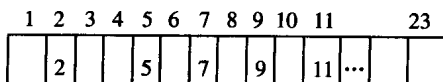


图 1-12 散列存储

## 7. 数据类型 (Data Type)

数据类型是一组具有相同性质的操作对象以及该组操作对象上的运算方法的集合。如整数类型、字符类型等，每一种数据类型都有其自身特点的一组操作方法（即运算规则）。

## 8. 抽象数据类型 (Abstract Data Type)

抽象数据类型是指一个数学模型以及在该模型上定义的一套运算规则的集合。在对抽象数据类型进行描述时，要考虑到完整性和广泛性。完整性就是要能体现所描述的抽象数据类型的全部特性，广泛性就是所定义的抽象数据类型适用的对象要广。在大型程序设计和系统软件开发中，对抽象数据类型用得较多。

## 1.2 数据结构的重要性

### 1.2.1 数据结构理论的发展历史

《数据结构》最早作为一门独立的课程出现是在 1968 年,当时在美国一些大学的计算机专业中开设了这一门课程。在这之前,数据结构的一些内容只在别的课程中讲述,如在表处理语言中阐述。1968 年虽然《数据结构》作为一门课程在美国的一些大学开始讲授,并且列入了教学计划,但是对讲授的内容,讲授的深浅度等并未作具体要求。并且在当时对于“数据结构”这个概念并未有统一的解释。几乎等同于图论,特别是与树、表的理论相提并论。但是在后来,数据结构的概念又包含了网络、代数、集合论、关系等现在《离散数据结构》中讲述的内容。

由于计算机在处理数据信息时,不仅要考虑数据本身的一些内在数学特性,而且要考虑数据与数据的关系,即数据的逻辑结构,同时还要考虑数据在计算机内的存储方法和形式,即数据的物理存储结构。这就使得《数据结构》的研究范围得到进一步扩展。1968 年,美国的唐·欧·克努特教授开创了数据结构的最初体系,他所著的《程序设计技巧》第一卷《基本算法》是第一本较系统地阐述数据的逻辑结构和存储结构及其操作的著作。

从 20 世纪 60 年代末到 70 年代初,出现了大型程序,从软件工程的角度出发,为了提高程序的可维护性、可调试性等性能要求,程序和数据必须分离。有的人认为程序设计的实质是为问题选择一种较好的数据结构加上一种好的算法。著名瑞士计算机科学家沃思(N. Wirth)就高度概括出了“程序=数据结构+算法”的经典等式。数据结构的实践基础是程序设计技术。反过来,程序设计技术又改进和提高数据结构理论的发展水平。

随着数据库系统技术的不断提高,大型数据库程序设计越来越多,对于数据处理的技术和范围要求越来越高,因此在数据结构理论中又增加了文件管理(特别是大型文件)的内容。

今天,虽然数据结构理论已经比较成熟,但对于它在面向各个专业应用领域的研究还在不断发展中,如多维图形数据结构、动画数据结构、音频数据结构。特别是它在实际工作和生活中的应用。

### 1.2.2 数据结构在计算机课程体系中的重要地位

目前在我国,从教学计划来说,《数据结构》不仅是计算机专业教学计划中的核心课程之一,而且是其他非计算机专业的主要选修课之一。它是介于数学、计算机硬件和计算机软件三者之间的一门核心课程,是计算机系统软件和应用软件研制者的必修课程。因此在全国计算机等级考试(二级以上)、计算机信息等方向的全国硕士入学考试、计算机专业的专升本考试、全国程序员水平(资格)考试以及各类资格认证中,数据结构知识所占试题的分值比例都非常大。从而可以看出数据结构在计算机课程体系中的重要地位。

从课程体系来说,数据结构是一门综合性很强的专业基础课。数据结构理论发展至今,已经成为一门比较成熟的课程,数据结构理论的应用范围已经渗透到编译系统、操作系统、数据库、人工智能、信息科学、企业管理、系统工程、计算机辅助设计及其他信息管理的应用。因为操作系统、编译系统数据库技术和人工智能都要用到队列、堆栈、树、图、广义表等内容,因而数据结构是后续课程操作系统、编译系统等的基础,同时也是开发软件所必须

具备的知识。

### 1.3 算法评价

#### 1.3.1 算法的定义及表示

##### 1. 算法的定义

做任何事情都有一定的步骤。如坐地铁，首先要兑换硬币，然后用硬币买地铁票，再到地铁入口处验票进入站台，最后上地铁。这些步骤都是有顺序的，而且缺一不可。所以广义地说，算法就是为解决问题而采取的步骤和方法。在程序设计中，算法是在有限步骤内求解某一问题所使用的一组定义明确的指令序列。通俗点说，就是计算机解题的过程。每条指令表示一个或多个操作。在这个过程中，无论是形成解题思路还是编写程序，都是在实施某种算法。前者是算法实现的逻辑推理，后者是算法实现的具体操作。

##### 2. 算法的表示

为了表示一个算法，可以用多种不同的方法。常用的有自然语言、传统流程图、结构化流程图、N-S流程图、伪代码、计算机语言表示法等。

##### 1) 自然语言表示法

自然语言就是人们日常使用的语言，如汉语、英语等。用自然语言表示算法的优点是通俗易懂；缺点是文字冗长，容易出现“二义性”。因此只有非常简单的问题的算法才用这种方法。

##### 2) 流程图表示法

就是用图形的方式，用规定的各种图框表示不同的操作。美国国家标准协会 ANSI (American National Standard Institute) 所规定的一些常用符号已为世界通用的流程表示符号。流程图表示法又有传统流程图表示法和结构化流程图两种形式。现在主要用结构化流程图表示。如选择结构算法用流程图表示如图 1-13 所示。虚线框内是一个选择结构。通过判断框内的条件选择执行 A 框或 B 框。

##### 3) N-S 流程图表示法

1973 年美国学者 I.Nassi 和 B.Shneiderman 提出了一种新的流程图方式。在这种新的流程图中，完全取消了带箭头的流程线。全部算法放在一个矩形框内，该框内还可以包含一些从属框，取名为 N-S 流程图。如选择结构算法用 N-S 流程表示如图 1-14 所示。

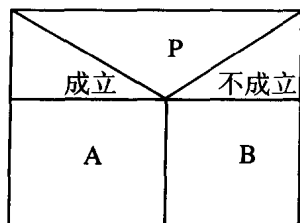
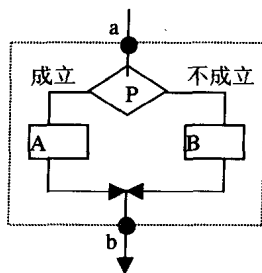


图 1-13 选择结构流程图

图 1-14 选择结构的 N-S 流程图

##### 4) 伪代码表示法

用一般流程图和 N-S 流程图表示算法都可以做到一目了然，直观易懂，但是费时费力，

而且修改麻烦。而用伪代码表示,虽然算法不如流程那样直观,也有可能出现逻辑上的错误,但用伪代码表示的算法书写较为自由,修改也容易实现,而且与用计算机语言表示的算法差别不是非常大,易于编写程序源代码。一般的专业软件设计者都习惯于用伪代码表示算法。

例如求  $1+2+3+\cdots+100$  的和,用伪代码表示其算法可写成如下形式:

```
开始
置 s 的初值为 0
置 i 的初值为 1
当  $i \leq 100$  时, 重复执行下面的累加操作:
```

```
s=s+i
i=i+1
( 执行累加操作的循环体到此结束 )
输出和 s 的值
结束
```

也可以用以下更接近程序设计语言的形式写出:

BEGIN(算法开始)

```
0:=s
l:=i
while ( $i \leq 100$ )
{ s+i  $\rightarrow$  s
  i+1  $\rightarrow$  i
}
```

```
print s
```

END(算法结束)

使用伪代码表示算法时需要注意一些书写规则:

(1) 在伪代码中,每一条指令占一行(if语句例外),指令后不跟任何符号(Pascal语言 and C语言中语句要以分号结尾)。

(2) 书写上的“缩进”表示程序中的分支程序结构。这种缩进风格也适用于 if-else 语句。用缩进表示程序的块结构可以大大提高代码的清晰性;同一模块的语句有相同的缩进量,次一级模块的语句相对与其父级模块的语句缩进。

例如:

```
line 1
line 2
    sub line 1
    sub line 2
        sub sub line 1
        sub sub line 2
    sub line 3
```

line 3

在 C 中这种关系用 { 和 } 的嵌套来表示:

```
line 1
line 2
{
    sub line 1
    sub line 2
    {
        sub sub line 1
    }
```