



21世纪高职高专计算机类立体化精品教材·基础课系列
工学结合教学改革与创新成果

数据结构(C/C++版)



姜君娜 安永丽 张立生 编著

教学资源包

电子课件: 包含10章PPT课件
教学参考: 包含课时规划和课程说明
课后习题: 课后习题答案
案例库: 任务相关教学案例
素材库: 任务相关素材
资源拓展: 包含学习网站和拓展阅读等
教学检测: 2份期末考试卷

教学服务: www.jxzy.com.cn



华南理工大学出版社
SOUTH CHINA UNIVERSITY OF TECHNOLOGY PRESS



21世纪高职高专计算机
工学结合教学改革与

教材·基础课系列

数据结构(C/C++版)



姜君娜 安永丽 张立生 编著



华南理工大学出版社
SOUTH CHINA UNIVERSITY OF TECHNOLOGY PRESS

·广州·

图书在版编目 (CIP) 数据

数据结构 (C/C++ 版) / 姜君娜, 安永丽, 张立生编著. —广州: 华南理工大学出版社, 2014. 7

21 世纪高职高专计算机类立体化精品教材. 基础课系列
ISBN 978-7-5623-4265-6

I. ①数… II. ①姜…②安…③张… 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 (2014) 第 115185 号

数据结构 (C/C++ 版)

SHUJU JIEGOU (C/C++ BAN)

姜君娜 安永丽 张立生 编著

出 版 人: 韩中伟

出版发行: 华南理工大学出版社

(广州五山华南理工大学 17 号楼, 邮编 510640)

http: //www. scutpress. com. cn E-mail: scutcl3@scut. edu. cn

营销部电话: 020-87113487 87111048 (传真)

项目策划: 王 磊

执行策划: 何丽云

责任编辑: 黄小虎 何丽云

印 刷 者: 北京市通县华龙印刷厂

开 本: 787mm × 1092mm 1/16 印张: 19 字数: 416 千

版 次: 2014 年 7 月第 1 版 2014 年 7 月第 1 次印刷

定 价: 39.00 元

版权所有 盗版必究 印装差错 负责调换

《数据结构（C/C++ 版）》针对采用 C 实现数据结构进行了全面的描述和深入的讨论。书中详细讨论了栈、队列、链表以及查找结构、高级树结构等功能，对斐波那契堆、伸展树、红黑树、2-3 树、2-3-4 树、二项堆、最小 - 最大堆、双端堆等新的数据结构进行了有效分析。书中对一些特殊形式的堆结构，诸如应用在双端优先队列中的最小 - 最大堆和双端堆的数据结构以及左高树、斐波那契堆、二项堆等数据结构，提供了经过编译和测试的程序和算法。书中还深入阐述了平摊复杂性问题，对大多数算法进行了时间复杂性的分析。

本书在编排上，注重理论与实践相结合，采用案例教学模式，突出实践环节，设置了任务描述、任务分析、准备知识、任务实施、知识链接、操作技巧、拓展提高、项目小结和项目考核等特色模块，意在提高学生的学习兴趣，促进学生的全面发展。

本书可作为高等院校计算机及相关专业的教材，也可作为成人教育和在职人员的培训教材，还可作为广大计算机爱好者的参考读物。

数据结构是计算机程序设计的重要理论技术基础课程。本书作为面向高等学校计算机相关专业学生而编写的教材，系统地介绍了数据结构的基础知识和应用。本书从逻辑结构和物理存储结构两个方面详细介绍了数据存储的原理和方法，根据各种存储结构的特点介绍了大量常用的计算方法。对于结构相对简单而又常见的线性结构，本书列举了许多实例，力求全面、细致地体现线性结构的优点和缺点。为了使线性链表这种比较抽象的数据结构易于理解，本书的例题都力求具体化，还添加了形象的插图。在介绍树和图等非线性数据结构时，在阐明概念的基础上增加了实用性和趣味性较强的例子。在介绍排序和查找等研究计算方法的内容时，不但给出了用类 C 语言表示的算法，对于较复杂的算法还给出了源程序。本书还对变量含义、设计思想和使用到的其他数据结构进行了详细说明。此外，还对许多算法执行过程中数据的变化进行了说明，体现了“化复杂为简单、化抽象为具体”的教学原则。

本书有如下特点：

- 涵盖教学大纲内容，兼顾学科知识的广度和深度，适用面广；
- 注重理论与实践相结合，采用任务教学模式，突出实践环节；
- 引入抽象数据类型的基本概念，有助于培养学生的数据抽象和算法设计能力；
- 以 C 伪码语言描述存储结构和算法，有助于提高学生的程序设计能力；
- 对算法进行详尽的定性或定量的时间分析，有助于奠定学生的算法分析基础。

本书的编写注意在学习理论知识的基础上重点培养学生的实际操作能力，通过一系列实例分析、实践等环节的训练，提高学生的实际应用能力。在编排上，本书设置了任务描述、任务分析、准备知识、任务实施、知识链接、操作技巧、拓展提高、项目小结和项目考核等特色模块，意在提高学生的学习兴趣，促进学生的全面发展。

本书由河北联合大学姜君娜、安永丽、张立生编写，其中项目一～项目五由姜君娜编写，项目六～项目八由安永丽编写，项目九由张立生编写。参与本书整理及校对工作的还有吴涛、阚连合、张航、李伟、封超、刘博、王秀华、薛贵军、周振江、张海兵、刘阁、刘翀、陈海彬等，在此一并表示感谢。

由于编者水平有限，书中欠妥之处，敬请广大读者批评指正。

编 者



CONTENTS

目 录

项目一 数据结构基本知识

任务：建立数学模型..... 2

任务描述..... 2

任务分析..... 2

准备知识..... 2

 1. 数据结构概述..... 2

 2. 基本术语..... 3

 3. 数据的存储结构..... 6

 4. 算法及算法分析..... 8

 5. C 语言预备知识..... 14

任务实施..... 21

项目小结..... 23

项目考核..... 23

项目二 线性表及其应用

任务一：利用顺序表对数据进行
 升序排序..... 26

任务描述..... 26

任务分析..... 26

准备知识..... 26

 1. 线性表概述..... 26

 2. 线性表的抽象数据类型和基本操作..... 27

 3. 线性表的顺序存储结构..... 31

任务实施..... 41

任务二：使用单链表对数据进行
 升序排序..... 42

任务描述..... 42

任务分析..... 42

准备知识..... 43

 1. 单链表..... 43

 2. 双向链表..... 53

 3. 循环链表..... 59

任务实施..... 59

项目小结..... 61

项目考核..... 61

项目三 栈的使用

任务：利用栈判断括号匹配情况..... 64

任务描述..... 64

任务分析..... 64

准备知识..... 64

 1. 栈的概述..... 64

2. 栈的抽象数据类型和基本操作.....	64
3. 栈的顺序存储结构.....	65
4. 栈的链式存储结构.....	69
5. 递归的概述.....	71
6. 河内塔 (Hanoi tower) 问题.....	72
任务实施.....	74
项目小结.....	78
项目考核.....	78

项目四 队列及其应用

任务：使用队列模拟打印机打印顺序.....	80
任务描述.....	80
任务分析.....	80
准备知识.....	80
1. 队列的概述.....	80
2. 队列的抽象数据类型和基本操作.....	81
3. 队列的顺序存储结构.....	82
4. 顺序队列的改进——循环队列.....	86
5. 队列的链式存储结构.....	88
任务实施.....	93
项目小结.....	96
项目考核.....	96

项目五 字符串及其应用

任务：使用字符串统计英文单词.....	98
任务描述.....	98
任务分析.....	98
准备知识.....	98
1. 字符串的概述.....	98
2. 字符串的抽象数据类型和基本操作.....	99
3. 字符串的顺序存储结构.....	101
4. 字符串的堆存储结构.....	106

5. 字符串的块链存储结构.....	109
任务实施.....	111
项目小结.....	112
项目考核.....	112

项目六 树及二叉树

任务一：计算二叉树的前序

遍历序列.....	114
任务描述.....	114
任务分析.....	114
准备知识.....	114
1. 树结构.....	114
2. 树形结构的种类.....	115
3. 树的相关术语.....	115
4. 二叉树的概述.....	116
5. 满二叉树和完全二叉树.....	116
6. 二叉树的性质.....	117
7. 二叉树的抽象数据类型和基本操作.....	118
8. 顺序存储结构.....	119
9. 链式存储结构.....	120
10. 二叉树的遍历.....	125
11. 线索二叉树的概述.....	130
12. 中序线索二叉树的构造和遍历.....	131
任务实施.....	135

任务二：使用 Huffman 树编写

C 语言程序.....	138
任务描述.....	138
任务分析.....	138
准备知识.....	138
1. 霍夫曼树相关术语.....	139
2. 霍夫曼树构造.....	140
任务实施.....	140

项目小结	143
项目考核	143

项目七 树、森林及二叉树

任务：实现深度为 3 的树	146
任务描述	146
任务分析	146
准备知识	146
1. 树的概述	146
2. 树和二叉树的三个主要差别	147
3. 森林的概述	147
4. 树的抽象数据类型和基本操作	147
5. 树的遍历	148
6. 森林的遍历	149
7. 树的存储结构	149
8. 树与二叉树的相互转换	152
9. 森林与二叉树的相互转换	153
10. K 叉树	153
任务实施	154
项目小结	161
项目考核	161

项目八 图形结构

任务：实现图的遍历	164
任务描述	164
任务分析	164
准备知识	164
1. 图的定义与基本术语	164
2. 图的概述	164
3. 图的相关术语	165
4. 图的抽象数据类型和基本操作	167
5. 邻接矩阵表示法	168

6. 邻接表表示法	174
7. 十字链表	181
8. 邻接多重表	183
9. 深度优先搜索	186
10. 广度优先搜索	189
11. 最短路径问题	193
12. 单源最短路径问题	194
13. 狄克斯特拉 (Dijkstra) 算法	195
14. 最小生成树	199
15. 最小生成树的性质	199
16. 构造最小生成树的算法	199
17. 拓扑排序	210
18. AOV 网	210
19. 拓扑排序 (Topological Sort)	211
任务实施	214
项目小结	218
项目考核	218

项目九 排序

任务：使用选择排序输出结果	220
任务描述	220
任务分析	220
准备知识	220
1. 排序的概述	220
2. 排序分类	221
3. 直接插入排序	221
4. 冒泡排序	224
5. 直接选择排序	226
6. 简单排序算法的时间代价对比	227
7. Shell 排序	227
8. 快速排序	230
9. 归并排序	234
10. 堆排序	239

11. 多关键码排序	244
12. 链式基数排序	244
任务实施	247
项目小结	249
项目考核	249

项目十 查找

任务：使用哈希法输出最大值	252
任务描述	252
任务分析	252
准备知识	252
1. 查找的基本概念	252
2. 顺序查找	254
3. 二分查找	258

4. 分块查找	261
5. 二叉排序树	264
6. 平衡二叉排序树	270
7. B 树和 B+ 树	271
8. 哈希表的概述	275
9. 哈希函数的构造方法	276
10. 处理冲突的方法	278
11. 哈希表的实现	280
12. 哈希表的查找分析	284
任务实施	285
项目小结	292
项目考核	292
参考文献	294

项目一

数据结构基本知识



项目要点

- 数据结构
- 数据存储结构
- 算法及算法分析
- C语言基础



引言

在本项目中，通过1个工作任务，向读者展示什么是数据结构，从而展开对数据结构、数据存储、算法等知识的讲解，为后续项目的完成打下坚实基础。

任务：建立数学模型



任务描述

使用数据结构知识，为企业员工信息管理、组织机构管理、学校排课等问题建立数学模型，从而实现使用计算机解决这些问题。



任务分析

建立数学模型的实质是对现实事物和问题的分析，从中提取出对象以及对象之间特有的关系。计算机能处理的数据也不再是简单的数值，而是字符串、图形、图像和语音等复杂数据。这些复杂数据不仅量大，而且具有一定的结构。数据结构所研究的就是这些有结构的数据。



准备知识

“数据结构”作为一门独立的课程在国外是从 1968 年才开始设立的。美国唐·欧·克努特教授在 1968 年开创了数据结构的最初体系。“数据结构”在计算机科学中是一门综合性的专业基础课，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。数据结构是计算机各专业必修的核心课程，是研究计算机程序设计的重要理论和技术的专业基础课程。

1. 数据结构概述

随着电子技术的发展，计算机逐渐深入到商业、制造业等人类社会各个领域，从而广泛地应用于数据处理和过程控制。与此相应，计算机能处理的数据也不再是简单的数值，而是字符串、图形、图像和语音等复杂数据。这些复杂数据不仅量大，而且具有一定的结构。数据结构所研究的就是这些有结构的数据，是研究数据对象的特征以及数据的组织和处理方式。在计算机科学中，数据结构（Data Structure）是计算机中存储、组织数据的方式，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。通常情况下，精心选择的数据结构可以带来最优效率（Algorithmic Efficiency）的算法，数据结构往往同高效的检索算法和索引技术有关。



知识链接

一般而言，数据结构的选择首先会从抽象数据类型的选择开始。一个设计良好的数据结构，应该在尽可能使用较少的时间与空间资源的前提下，为各种临界状态下的运行提供支持。数据结构可通过编程语言所提供的数据类型、引用（Reference）及其他操作加以实现。不同种类的数据结构适合于不同种类的应用，而部分甚至专门用于特定的作业任务。例如，当计算机网络依赖于路由表运作时，B 树高度适用于数据库的封装。

在许多类型的程序设计中,选择适当的数据结构是一个主要的考虑因素。许多大型系统的构造经验表明,封装的困难程度与最终成果的质量与表现,都取决于是否选择了最优的数据结构。在许多时候,确定了数据结构后便能很容易地得到算法。而有些时候,方向则会颠倒过来:例如当某个关键作业需要特定数据结构下的算法时,会反过来确定其所使用的数据结构。然而,不管是哪种情况,数据结构的选择都是至关重要的。

2. 基本术语

数据 (Data): 计算机中的数据是广义的,包括了数、字符、字符串、表、文件等,声音、图形、图像都属于数据的范畴。数据指计算机加工的“原料”,是对客观事物采用计算机能够识别、存储和处理的形式所进行的描述。数据就是计算机化的信息,是计算机中符号化的特定表示形式。

数据元素 (Data Element): 是数据的基本单位,是数据集合的个体,在计算机中通常作为一个整体进行考虑和处理。一个数据元素可由若干个数据项组成。此时的数据元素通常称为记录 (Record)。比如例 1-1 中描述的某一条员工信息。

数据项 (Data Item): 是不可分割的、含有独立意义的最小数据单位,数据项有时也称为字段 (Field) 或域。如员工记录中的姓名、性别等。

数据对象 (Data Object): 是性质相同的数据元素的集合。它是数据的一个子集。例如,整型数据对象是集合 $\{0, \pm 1, \pm 2, \dots\}$, 字符数据对象是 $\{a, b, c, \dots\}$ 等,数据对象可以是有限的,也可以是无限的。

数据结构 (Data Structure): 是相互之间存在一种或多种关系的数据元素的集合。例如表 1-1 员工基本信息中数据元素集合是员工记录集,而数据元素集合上的关系就是它们在员工基本信息表中的前驱和后继的关系,这种数据之间的关系是一对一的,是线性的。数据元素相互之间的关系称为结构 (Structure)。

数据类型 (Data Type): 是高级程序设计语言中的概念,是数据的取值范围和对数据进行操作的总和。

例如 C 语言中的整型及定义在其上的一组操作 (加、减、乘、除等)。数据类型是一组性质相同的值集合以及定义在这个值集合上的一组操作的总称。数据类型中定义了两个集合,即类型的取值范围和该类型可允许使用的一组运算。例如高级语言中的整型、实型、字符型就是已经实现的数据结构的实例。从这个意义上讲,数据类型是高级语言中允许的变量种类,是程序语言中已经实现的数据结构 (即程序中允许出现的数据形式)。



拓展提高

在 C 语言中,整型类型 (int) 可能的取值范围是 $-32768 \sim +32767$, 可用的运算符集合为加、减、乘、除、取模 (+、-、*、/、%)。从硬件的角度来看,它们的实现涉及“字”、“字节”、“位”、“位运算”等;从用户的观点来看,并不需要了解整数在计算机内是如何表示、运算细节是如何实现的,用户只需要了解整数运算的外部运算特性,而不必了解整数在计算机内部运算的细节,就可运用高级语言进行程序设计。

chapter
01chapter
02chapter
03chapter
04chapter
05chapter
06chapter
07chapter
08chapter
09chapter
10

抽象数据类型 (Abstract Data Type, ADT): 是指一些数据以及对这些数据所进行的操作的集合。数据之间有一定的关系, 不同的抽象数据类型的数据之间的关系不同。对数据进行的操作由数据间的关系决定。数据间的关系不同, 对数据的操作也就不同。这些操作既向程序的其余部分描述了这些数据是怎么样的, 也允许程序的其余部分改变这些数据。一个 ADT 可能是一个图形窗体以及所有能影响到该窗体的操作, 也可以是一个文件以及对这个文件进行的操作, 或者是一张保险费率表及相关操作等。

抽象数据类型是基于一类逻辑关系的数据类型以及定义在这个类型之上的一组操作。抽象数据类型的定义取决于客观存在的一组逻辑特性, 而与其在计算机内如何表示和实现无关, 即不论其内部结构如何变化, 只要它的数学特性不变, 都不影响其外部使用。从某种意义上讲, 抽象数据类型和数据类型实质上是一个概念。整数类型就是一个简单的抽象数据类型实例。“抽象”的意义在于数学特性的抽象。一个抽象数据类型定义了一个数据对象, 数据对象中各元素间的结构关系, 以及一组处理数据的操作。



知识链接

抽象数据类型通常是指由用户定义用以表示应用问题的数据模型, 通常由基本的数据类型组成, 并包括一组相关的操作。

抽象数据类型包括定义和实现两方面, 其中定义是独立于实现的。定义仅给出一个抽象数据类型的逻辑特性, 不必考虑如何在计算机中实现。抽象数据类型的特征是使用与实现分离, 实现封装和信息隐蔽, 也就是说, 在抽象数据类型设计时, 类型的定义与其实现分离。另外, 抽象数据类型的含义更广, 不仅限于各种不同的计算机处理器中已定义并实现的数据类型, 还包括设计软件系统时用户自己定义的复杂数据类型。所定义的数据类型的抽象层次越高, 含有该抽象数据类型的软件复用程度就越高。抽象数据类型定义该抽象数据类型需要包含哪些信息, 并根据功能确定公共界面的服务, 用户可以使用公共界面中的服务对该抽象数据类型进行操作。从用户的角度来看, 只要了解该抽象数据类型的规格说明, 就可以利用其公用界面中的服务来使用这个类型, 不必关心其物理实现, 从而集中考虑如何解决实际问题。

抽象数据类型是近年来计算机科学中提出的最重要的概念之一, 它集中体现了程序设计中一些最基本的原则: 分解、抽象和信息隐藏。严格地可以用代数系统形式定义一个抽象数据类型, 可以把抽象数据类型看成是定义了一组运算的数学模型。

ADT 的定义采用下述格式:

```
ADT<ADT 名>
{
    数据对象: <数据对象的定义>
    数据关系: <结构关系的定义>
    基本操作: <基本操作的定义>
} ADT<ADT 名>
```

其中数据对象和结构关系的定义采用数学符号和自然语言描述，而基本操作的定义格式为 C 语言基本函数的定义形式：

```

函数类型  函数名 ([ 参数列表 ])
[ 类型定义列表 ]
{
    函数体
}
    
```

参数表中的参数有两种：第一种参数只为操作提供待处理数据，又称值参；第二种参数既能为操作提供待处理数据，又能返回操作结果，也称变量参数。操作前提描述了操作执行之前数据结构和参数应满足的条件，操作结果描述操作执行之后，数据结构的变化状况和应返回的结果。抽象数据类型可用现有计算机语言中已有的数据类型，即固有数据类型来表示和实现。

在“组织机构管理问题”中描述的数据元素集合是组织机构的集合，而组织机构之间的关系就是数据元素之间的关系。这种数据元素相互之间的关系也是一种结构，是层次结构或称树状结构。

数据结构的定义形式为：

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

其中 D 是数据元素的有限集； S 是 D 上关系的有限集。

根据数据元素之间关系的不同特性，通常有下列 4 类基本的数据结构，如图 1-1 所示。

①集合 (Set)：该结构中的数据元素除了存在“同属于一个集合”的关系外，不存在任何其他关系，如图 1-1a 所示。

②线性结构 (Linear Structure)：该结构中的数据元素存在着一对一的关系，如图 1-1b 所示。

③树形结构 (Tree Structure)：该结构中的数据元素存在着一对多的关系，如图 1-1c 所示。

④图状结构 (Graphic Structure)：该结构中的数据元素存在着多对多的关系，如图 1-1d 所示。

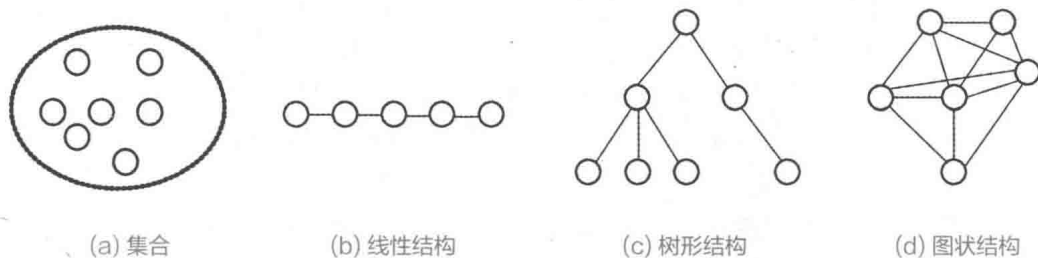


图 1-1 基本的数据结构

有时也称树状结构和图状结构为非线性结构。

chapter
01

chapter
02

chapter
03

chapter
04

chapter
05

chapter
06

chapter
07

chapter
08

chapter
09

chapter
10

某课题小组中有一位教授,教授可以带 1~3 名研究生,每名研究生可以带 1~2 名本科生。设计课题小组的数据结构。

设计课题小组的数据结构为: $\text{Group} = (P, R)$

其中, P 表示数据对象; R 表示数据对象中数据元素的关系。

P 、 R 的描述如下:

$P = \{T, G_1, \dots, G_n, S_{11}, \dots, S_{nm}\} (1 \leq n \leq 3, 1 \leq m \leq 2)$

$R = \{R_1, R_2\}$

$R_1 = \{ \langle T, G_i \rangle \mid 1 \leq i \leq n, 1 \leq n \leq 3 \}$

$R_2 = \{ \langle G_i, S_{ij} \rangle \mid 1 \leq i \leq n, 1 \leq j \leq m, 1 \leq n \leq 3, 1 \leq m \leq 2 \}$

其中, T 表示教授; G 表示研究生; S 表示大学生。

严格地说,数据结构包括两方面的内容:数据的逻辑结构(Logical Structure)和数据的物理结构(Physical Structure)。 Group 定义的数据结构是对操作对象的一种数学描述,其中“关系”是描述数据元素之间的逻辑关系,因此称为数据的逻辑结构。与逻辑结构相对应的是数据的物理结构。数据的物理结构又称为数据的存储结构,是指数据的逻辑结构在计算机中的映像(又称表示),即数据结构在计算机中的存储方法。由于数据结构包括数据元素集合及数据元素之间的关系,所以数据的存储结构也应该包含这两部分内容,即包括数据元素的映像和元素间关系的映像两部分。数据元素之间的关系在计算机中有两种不同的表示方法,即顺序映像和非顺序映像,并由此得到两种不同的存储结构,即顺序存储结构和链式存储结构。顺序存储结构的特点是借助于元素在连续空间的存储器中的相对位置来表示数据元素之间的逻辑关系。非顺序映像的特点是借助于指示元素存储地址的指针(Pointer)来表示数据元素之间的逻辑关系。



知识链接

任何一个程序都涉及数据的存储结构。因为对于逻辑结构上的同一种运算,其具体的实现方法依存储结构的不同而变化。所以,当描述问题的模型确定之后,首要问题就是确定其存储结构。

3. 数据的存储结构

数据的存储结构是建立一种由逻辑结构到存储空间的映射:对于逻辑结构 (K, r) , 其中 $r \in R$, 对它的结点集合 K 建立一个从 K 到存储器 M 的单元的映射: $K \rightarrow M$, 其中每个结点 $j \in K$ 都对应一个唯一的连续存储区域 $c \in M$ 。

常用的数据存储结构有两类:一类是顺序存储结构,一类是链式存储结构。

(1) 顺序存储结构

顺序存储结构是指逻辑上相邻的数据元素,其结点的物理位置也相邻,数据元素之间的关系由结点的邻接关系体现。顺序存储把一组结点存放在按地址相邻的存储单元里,结点间的逻辑关系用存储单元的自然顺序关系来表达的,即用一块存储区域存储线性数据结构。顺序存储法为使用整数编码访问数据结点提供了便利。因为,顺序

存储方法的存储空间除了存储有用数据外,没有用于存储其他附加的信息,所以顺序存储结构一般也被称为紧凑存储结构。计算机的内存单元是一维结构,这种存储结构很方便实现。

比如在本项目“企业员工信息管理问题”的题目中给出的员工信息逻辑结构,这个结构是线性的,各个员工记录前后是物理相邻的。

顺序存储结构有下列特点:

①结点中只存放数据元素本身的信息,无附加内容。

②由于每个结点所占存储空间大小一致,并且按顺序存储,所以只要知道第1个结点的地址,就可以通过计算直接确定结构中第 i 个结点的地址,从而直接存取第 i 个数据元素。

③由于可根据公式直接确定第 i 个结点的地址而直接存取第 i 个元素,所以数据元素的存取操作速度较快。

④插入、删除数据元素时,由于需要保持数据元素之间的逻辑关系,必须移动大量元素,因此实现起来较慢。关于这一点,将在第2章介绍线性表时详细说明。

⑤顺序存储结构有两种方式分配空间,一种是静态结构,另一种动态结构。当用静态结构时,存储空间一旦分配完毕,其大小就难以改变。因此,当表中元素个数难以估计时,分配的空间大小也就难以确定。预分配的空间太大会造成浪费,空间太小,又可能发生存放不下的溢出现象。

(2) 链式存储结构

链式存储是在结点的存储结构中附加指针字段来存储结点间的逻辑关系。链式存储中数据结点包括两部分:数据字段存放结点本身的数据,指针字段存放指向其后继结点的指针。链式存储方法适用于那些需要经常进行增删结点的复杂数据结构。

在链式存储结构中,逻辑上相邻的数据元素,其结点的物理位置不一定相邻,因此结点之间是否邻接并不能反映数据元素的逻辑顺序。在这种情况下,存储结构要反映数据元素在逻辑结构中的关系,只有在结点中增加信息来指明与其他结点之间的这种关系,这个增加的信息就是指针。前一个结点的指针指向后一个结点,多个结点的指针一起形成一个链,因此称这种存储结构为链式存储结构。链式存储结构既可用于实现线性数据结构,也可用于实现非线性数据结构。对于非线性数据结构来说,每个数据元素在逻辑上可能与多个数据元素相邻,而计算机内存的一维地址结构限制了每一个结点只能与前、后各一个结点相邻,结点的相邻反映不出一个元素与多个元素的相邻关系,所以非线性逻辑结构只能用链式存储结构来实现。树、图等就是这种非线性数据结构。在链式存储结构的每个结点中,数据域可分为两类:一类用于存放数据元素本身的信息,称作信息域;另一类用于存放指针,称作指针域。

链式存储结构的特点主要有以下几方面。

①结点中除存放数据元素本身的信息外,还需存放附加的指针。

②不能直接确定第 i 个结点的存储位置。要存取第 i 个结点的信息,必须从第1个结点开始查找,沿指针顺序取出第 $i-1$ 个结点的指针域,再取出第 i 个结点的信息,

chapter
01chapter
02chapter
03chapter
04chapter
05chapter
06chapter
07chapter
08chapter
09chapter
10