

大学本科计算机专业应用型规划教材

丛书主编：高林

数据结构

(C++ 版)

陈明 编著

清华大学出版社



大学本科计算机专业应用型规划教材

丛书主编：高林

（封面设计）

数据结构

（C++ 版）

陈明 编著

清华大学出版社
北京

内 容 简 介

本书系统地介绍了各种典型的数据结构,主要包括:线性表、栈和队列、串、数组和广义表、树、图、查找、排序、递归和文件。为了加强对算法的理解,还介绍了算法分析方面的内容。本书叙述精练、概念清楚、注重实用、逻辑性强,各章中所涉及的数据结构与算法都给出了 C++ 语言描述,并附有大量的习题,便于学生理解与掌握。

本书可作为高等院校计算机专业及相关专业的教材,也可作为计算机应用技术人员的参考书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用清华大学核研院专有核径迹膜防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

数据结构(C++版)/陈明编著. —北京:清华大学出版社,2005.3

(大学本科计算机专业应用型规划教材/高林主编)

ISBN 7-302-10186-8

I. 数… II. 陈… III. 数据结构—高等学校—教材 IV. TP311.12

中国版本图书馆 CIP 数据核字(2004)第 137983 号

出 版 者: 清华大学出版社

<http://www.tup.com.cn>

社 总 机: 010-62770175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

组稿编辑: 谢 琛

文稿编辑: 汪汉友

印 刷 者: 北京季峰印刷有限公司

装 订 者: 三河市李旗庄少明装订厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 24.25 字数: 571 千字

版 次: 2005 年 3 月第 1 版 2005 年 3 月第 1 次印刷

书 号: ISBN 7-302-10186-8/TP·6965

印 数: 1~5000

定 价: 29.00 元

大学本科 计算机专业应用型规划教材

编 委 会

主 编：高 林

副 主 编：王 利 鲍 洁

委 员：（按姓氏笔画为序）

王宝智	古 辉	孙悦红	安淑芝
肖 刚	陈 明	张 玲	张建忠
周海燕	赵乃真	娄不夜	顾巧论
崔武子	鲍有文		

策划编辑：谢 琛 汪汉友

丛书序

大学本科计算机专业应用型规划教材

为适应我国“以信息业带动工业化,发挥后发优势,实现社会生产力的跨越式发展”以及大力发展制造业和优化产业结构的要求,应用型人才培养已成为高等学校人才培养的重要任务。

以微电子技术为基础、计算机技术为主体的信息技术,是当前人类社会中发展最快、渗透性最强、应用面最广的先导技术。信息技术的广泛应用推动着以信息产品制造业、软件业、信息系统集成业和信息咨询服务业为主体的信息产业的发展。在新的世纪里,信息已成为重要的生产要素和战略资源,信息技术成为先进生产力的代表,信息产业将发展成为现代产业的带头产业,人类即将跨越工业时代进入信息时代。因此,信息化成为当今世界经济和社会的发展趋势,大力推进社会和国民经济信息化是推进我国社会主义现代化建设的重要任务。计算机和信息技术的发展不仅需要大批专业技术人才,而且还产生了一批新的职业岗位,毋庸置疑,信息及其相关职业将成为未来最紧缺的职业。

计算机和信息技术与应用的人才需求将呈多元化、多层次趋势,表现在科学、技术、产业、应用、服务诸多方面。不仅需要从事科学、技术研发的人才,而且更需要把研发成果转变为现实产品的技术和管理人才,不仅要有能从事计算机和信息科学、技术工作的人才,而且更需要能从事计算机和信息产业、应用、服务工作的人才,以及在各类人才中的精英人才、领军人物。这实际是对我国计算机和信息类高等教育改革提出了新的要求和新的课题,要求我国高等教育进行结构调整,满足人才培养的多元化,大力培养具有计算机和信息技术专长的应用型人才——他们是这些领域的技术专家和管理专家,可以在相应的行业、企业担任各种技术工作。

目前,我国高等教育中应用型人才培养模式相对落后,如何发展应用型教育已成为课程改革的主要任务。本套教材是以培养计算机和信息类专业本科应用型人才为目的进行的课程与教材改革尝试。在本套教材的策划过程中,清华大学出版社多次组织了由行业企业专家和有丰富教学经验的一线教师参加的研讨会,对应用型高等教育的规律和在计算机教学中的体现进行了深入的研讨。在此基础上我们力求能从整体上把握计算机和信息类应用型人才培养的特征,并体现在这套教材的编写过程中:在教材编写的指导思想上,力求在保持学科科学性的同时,体现工程和技术学科的系统性;在教材

的内容组织上,尽量采用以问题为中心的写作方法,加强案例性教学;在理论联系实际和加强能力培养方面,增加方案性设计习题和实际训练性题目,以培养学生的专门技术能力和完成实际工作任务的能力。

计算机和信息类应用型教材编写还处于改革的初步尝试阶段,希望使用这套教材的教师也能够参与到教材建设工作中来,并提出宝贵意见,以便推动课程改革并提高教材质量。

高 林

2004 年 5 月

目 录

数据结构(C++版)

第 1 章 绪论	1
1.1 数据结构的重要性	1
1.2 面向对象程序设计	2
1.2.1 面向对象程序设计方法	2
1.2.2 C++ 的特征及基本概念	3
1.3 基本术语	4
1.4 抽象数据类型	6
1.5 数据结构的概念	8
1.6 数据的逻辑结构	10
1.7 数据的存储结构	11
1.8 数据的运算	13
1.9 数据的逻辑结构、存储结构及数据的运算的关系	14
1.10 算法的描述	14
小结	15
习题	15
第 2 章 算法分析	16
2.1 算法分析的概念	16
2.2 算法运行时间举例	18
2.3 最大连续子序列之和的问题	18
2.3.1 简单易懂的 $O(n^3)$ 算法	19
2.3.2 一个改进的 $O(n^2)$ 算法	20
2.3.3 一个线性算法	21
2.4 静态搜索问题	23
2.4.1 顺序搜索	23
2.4.2 二分搜索	24
2.4.3 插值搜索	26
2.5 检验一个算法分析	26
2.6 Big-Oh 分析法的限制	27
小结	28

习题	28
第 3 章 线性表	30
3.1 线性表及其抽象数据类型说明	30
3.1.1 线性表及其逻辑结构	30
3.1.2 线性表的抽象数据类型描述	34
3.2 线性表的顺序存储	35
3.2.1 顺序存储	35
3.2.2 顺序表类的定义	36
3.2.3 顺序表类的实现	36
3.3 线性表的链式存储	40
3.3.1 线性链表的存储结构	40
3.3.2 线性链表类的定义	41
3.3.3 线性链表类的实现	43
3.3.4 循环链表	47
3.3.5 循环链表类的实现	48
3.3.6 双向链表	49
3.3.7 可利用空间表	51
3.3.8 表遍历器	52
3.4 线性表的顺序存储和链式存储的比较	54
3.5 链式存储结构的应用	55
3.5.1 约瑟夫问题	55
3.5.2 一元多项式求和	56
小结	60
习题	60
第 4 章 栈和队列	63
4.1 栈	63
4.1.1 栈的定义	63
4.1.2 栈的顺序存储结构	66
4.1.3 栈的链式存储结构	69
4.1.4 顺序栈和链式栈的比较	72
4.2 栈的应用	72
4.2.1 迷宫问题	72
4.2.2 表达式求值	75
4.2.3 汉诺塔问题	78
4.2.4 数制转换	80
4.2.5 行编辑	81

4.3	队列	82
4.3.1	队列的定义	82
4.3.2	队列的顺序存储	85
4.3.3	队列的链式存储	92
4.3.4	顺序队列与链式队列的比较	96
4.3.5	优先队列	96
4.4	队列的应用	97
4.4.1	解决设备速度不匹配问题	97
4.4.2	舞伴问题	97
4.4.3	火车车厢重排	99
	小结	101
	习题	101
第5章	串	105
5.1	C++语言的字符和字符串	105
5.2	串的基本概念	106
5.3	串的存储结构	107
5.3.1	串的顺序存储结构	107
5.3.2	串的链式存储结构	109
5.3.3	串的索引存储结构	110
5.4	串的操作	111
5.4.1	常用的C++字符串函数	111
5.4.2	串的抽象数据类型的描述	112
5.4.3	串的定义	113
5.4.4	部分成员函数的实现	114
5.5	串的基本运算与实现	117
5.5.1	串插入	117
5.5.2	串删除	118
5.6	模式匹配	120
5.6.1	模式匹配的BF算法	120
5.6.2	模式匹配的KMP算法	122
5.7	串在文本编辑中的应用	126
	小结	128
	习题	128
第6章	数组和广义表	130
6.1	C++中数组的定义及抽象数据类型表示	130
6.1.1	C++中数组的定义	130

6.1.2 数组的抽象数据类型表示·····	131
6.2 数组的顺序存储结构·····	132
6.3 矩阵的压缩存储·····	133
6.3.1 特殊矩阵的压缩存储·····	134
6.3.2 稀疏矩阵的压缩存储·····	136
6.4 广义表的概念·····	143
6.5 广义表的存储结构表示·····	144
6.6 广义表的运算·····	146
小结·····	153
习题·····	153
第7章 树 ·····	157
7.1 树的基本概念·····	157
7.1.1 树的定义·····	157
7.1.2 树的表示形式·····	158
7.1.3 树的常用术语·····	159
7.1.4 树的基本操作·····	159
7.1.5 一个树的接口·····	160
7.1.6 树的基本算法·····	161
7.2 二叉树·····	162
7.2.1 二叉树的定义·····	162
7.2.2 二叉树的性质·····	164
7.2.3 二叉树的接口·····	166
7.2.4 二叉树的存储结构·····	167
7.2.5 二叉树的遍历·····	175
7.2.6 二叉树遍历的应用·····	178
7.3 线索二叉树·····	179
7.3.1 线索二叉树的类定义·····	179
7.3.2 中序线索二叉树·····	183
7.4 树、森林和二叉树的关系·····	186
7.4.1 树的存储结构·····	186
7.4.2 森林与二叉树的转换·····	189
7.4.3 树和森林的遍历·····	192
7.5 霍夫曼树及其应用·····	194
7.5.1 霍夫曼树的定义·····	194
7.5.2 霍夫曼树的构造·····	195
7.5.3 霍夫曼树在编码问题中的应用·····	198
小结·····	200

习题	200
第8章 图	204
8.1 图的基本概念	204
8.1.1 图的定义及基本概念	204
8.1.2 图的抽象数据类型	208
8.2 图的存储结构	209
8.2.1 邻接矩阵表示法	210
8.2.2 邻接表	215
8.2.3 十字链表	221
8.2.4 邻接多重表	223
8.3 图的遍历	225
8.3.1 深度优先搜索	225
8.3.2 广度优先搜索	227
8.3.3 欧拉回路	228
8.4 图的连通性	230
8.4.1 连通分量	230
8.4.2 重连通分量	232
8.5 生成树	233
8.5.1 普里姆算法	234
8.5.2 克鲁斯卡尔算法	237
8.6 最短路径	239
8.6.1 单源最短路径	240
8.6.2 求每一对顶点之间的最短路径	242
8.7 拓扑排序	243
8.8 关键路径	247
小结	253
习题	254
第9章 查找	258
9.1 基本概念	258
9.2 线性表的查找	259
9.2.1 顺序查找	259
9.2.2 折半查找	261
9.2.3 索引查找	263
9.2.4 分块查找	267
9.3 树表查找	269
9.3.1 二叉查找树	270

9.3.2	平衡二叉树	276
9.3.3	B ⁺ 树	281
9.4	哈希表的查找	283
9.4.1	哈希表	283
9.4.2	构造哈希表的基本方法	284
9.4.3	解决冲突的方法	286
9.4.4	哈希表的查找方法	289
9.5	各种查找方法的比较	290
	小结	291
	习题	291
第 10 章	排序	295
10.1	基本概念	295
10.2	内部排序	298
10.2.1	插入排序	298
10.2.2	交换排序	303
10.2.3	选择排序	307
10.2.4	归并排序	315
10.2.5	基数排序	319
10.3	内部排序方法比较	323
10.4	外部排序简介	325
	小结	325
	习题	326
第 11 章	递归	330
11.1	递归的定义	330
11.2	常见递归问题	331
11.2.1	汉诺塔问题	331
11.2.2	八皇后问题	332
11.2.3	表达式树	335
11.3	递归的实现	337
11.4	消除递归	340
11.4.1	尾递归和单向递归的消除	340
11.4.2	用栈模拟系统运行时的栈	341
11.5	递归的评估	344
	小结	345
	习题	346

第 12 章 文件	348
12.1 外存储器的介绍	348
12.2 磁盘	349
12.3 有关文件的概念	350
12.3.1 文件及其类别	351
12.3.2 文件的操作	352
12.4 文件的组织	353
12.4.1 顺序文件	354
12.4.2 索引文件	355
12.4.3 散列文件	361
12.4.4 多关键字文件	362
12.5 外部排序	365
12.5.1 外部排序的简单方法	365
12.5.2 两路归并	365
12.5.3 多路归并	368
12.6 文件的索引结构	369
12.6.1 索引向量	369
12.6.2 树形索引结构	370
小结	370
习题	370
参考文献	372

第1章

数据结构(C++版)

绪 论

在深入学习数据结构之前,首先了解一下学习数据结构的意义、数据结构及相关概念。这对于深刻理解本书后面章节的内容将会有很大的帮助。

1.1 数据结构的重要性

在计算机发展的初期,人们使用计算机主要是为了处理数值计算问题,程序设计人员也主要把精力集中在程序设计的技巧上,但是随着计算机应用领域的扩大和软、硬件的发展,计算机对信息的加工处理已从单一的数值计算发展到大量解决非数值计算问题,其加工处理的信息也由简单的数值发展到字符、图像、声音等具有复杂结构的数据。而数据结构就是伴随着计算机的产生和发展而成长起来的一门崭新的计算学科。

在非数值计算问题中,数据之间的相互关系一般无法完全用数学方程式加以描述,并且数据的表示方法和组织形式直接关系到程序对数据的处理效率,而系统程序和许多应用程序的规模很大,结构复杂,这时人们考虑问题的关键已不再是分析数据和计算方法,而是是否能设计出合适的数据结构,有效地解决问题。

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。这里面涉及到两个问题:信息的表示和信息的处理。其中,信息的表示和组成又直接关系到处理信息的程序的效率。通过例 1-1 就能看出一个合理的数据结构的重要性。

【例 1-1】 电话号码查询问题。

设计一个查询某市或某单位私人电话号码的程序。要求对任意给出的一个姓名,若有该人电话号码,则迅速找到;否则指出没有该人电话号码。

要解此问题首先构造一张电话号码登记表。表中每个结点存放两个数据项:姓名和电话号码。

要写出好的查找算法,取决于这张表的结构及存储方式。最简单的方式是将表中结点顺序地存储在计算机中。查找时从头开始依次查对姓名,直到找出正确的姓名或是找遍整个表均没有找到为止。这种查找算法对于一个不大的单位或许是可行的,但对一个有成千上万私人电话的城市就不实用了。若这张表是按姓氏排列的,则可另造一张姓氏索引表,采用如图 1-1 所示的存储结构。那么查找过程是先在索引表中查对姓氏,然后根

据索引表中的地址到电话号码登记表中核查姓名,这样查找登记表时就无需查找其他姓氏的名字了。因此,在这种新的结构上产生的查找算法就更为有效。

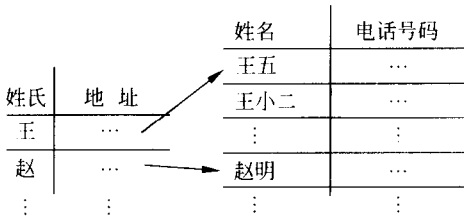


图 1-1

随着计算机的普及,信息量的增加,信息范围的拓宽,使得许多系统程序和应用程序的规模很大,且结构相当复杂。这就要求人们对计算机程序加工的对象进行系统地研究,即研究数据的特性以及数据之间存在的关系,而数据结构(data structure)正是描述数据的特性以及数据之间存在的关系的一门课程。它是计算机专业的核心课程之一,并在众多的计算机系统软件和应用软件中都要用到。可以这样说,数据结构不仅是一般程序设计的基础,而且是实现编译程序、操作系统、数据库系统及其他系统程序和大型应用程序的基础。因此,仅掌握几种计算机语言是难以应付众多复杂的研究课题的,要想有效地使用计算机,还必须学习数据结构的知识。

瑞士计算机科学家 N. Wirth 教授曾提出这样一个等式:

算法 + 数据结构 = 程序

这个等式形象的描述了算法、数据结构和程序之间的关系,这里的数据结构指的是数据的逻辑结构和存储结构,而算法就是对数据运算的描述。由此可见,程序设计的实质就是对实际问题选取一种优秀的数据结构,加之设计一个优秀的算法,而且优秀的算法很大程度上取决于描述实际问题的数据结构。对于一个程序设计人员来说,学好数据结构是十分重要的。

1.2 面向对象程序设计

1.2.1 面向对象程序设计方法

1. 面向对象

简单地说,面向对象是把一组相互无联系的对象有效地集成在一起的软件,而这些对象都是将数据结构和行为封装在一起的。这与传统的程序设计(将数据结构与行为分离)完全不同。

2. 面向对象程序设计方法的特征

面向对象程序设计方法是基于客观世界对象模型的软件开发方法。当设计和实现一个客观系统时,如能在满足需求的条件下,把系统设计成由一些相对固定的部分组成的最

小集合,则这个设计就是优秀的,而这些相对固定的部分就被看成是一些不同的对象。面向对象方法就是要建立与语言无关的对象模型,并围绕这些对象进行系统设计。

面向对象程序设计方法有如下基本特征。

(1) 对象是数据和操作的封装体,较好地实现了数据的抽象。

(2) 面向对象方法的继承性体现了概念分离抽象。下层对象继承上层对象的属性和操作,因而便于软件的演化和扩充。

(3) 面向对象方法具有信息隐藏性。对象将其实现的细节隐藏在内部,所以无论是对象功能的完善扩充,还是对象实现的修改,都不会对外界产生影响。这就保证了面向对象软件的可构造性和易维护性。

(4) 面向对象方法用消息将对象动态链接在一起。面向对象方法采用了灵活的消息传递方式,从而便于在概念上体现并行和分布式结构。

1.2.2 C++ 的特征及基本概念

面向对象语言 C++ 有如下重要特征:

- 类和数据封装;
- 构造函数和析构函数;
- 私有、公有和受保护;
- 对象和消息;
- 友元函数;
- 类中运算符和函数名重载;
- 派生类;
- 虚函数;
- 流库。

下面简单介绍一下 C++ 的一些基本概念。

(1) 类。类是一种数据结构,它把数据和处理数据的过程封装在一起。类和普通数据类型的主要区别在于类可以拥有成员数据和成员函数。

(2) 构造函数。构造函数是一种特殊的成员函数,用于在内存中建立具体的对象。构造函数必须初始化成员变量,申请必要的内存空间,并将内存转化成对象。一个类可以同时拥有几个带有不同参数的构造函数。

(3) 析构函数。析构函数与构造函数正好相反,它是一种当对象被取消时才被调用的特殊的成员函数。一个类只能有一个析构函数。析构函数和构造函数一样都没有返回类型和返回值。

(4) 私有(private)、公有(public)和受保护(protected)。每个类的成员变量和成员函数都有三种访问权限,即 private、public 和 protected。声明为 private 的成员变量和成员函数只能被该类的成员函数访问。声明为 public 的成员变量和成员函数,可以被其他类的成员函数访问。声明为 protected 的成员变量和成员函数不能被其他类的成员函数访问,但子类可以访问父类中声明为 protected 的成员函数。

(5) 对象和消息。对象是用发送的消息操作的,对象响应消息。每个消息都用类中

相应的方法给出。

(6) 重载。重载包含运算符重载和函数重载。运算符重载是指对已存在的运算符赋予新的含义。函数重载是指相同的函数名称,如果参数个数不同,或者是参数个数相同但类型不同,函数具有不同的功能。重载函数的意义在于它能用同一个函数名访问一组相关的函数。

(7) 继承。继承是在已有类的基础上定义新类的机制。继承又可以分为单继承和多重继承。单继承是指派生类继承自一个基类,而多重继承则是派生类继承自两个或两个以上的基类。

(8) 多态性。多态性是指程序在执行过程中可以保持多种不同类型的值。具有多态性的函数可以对多种不同类型的对象进行操作。

(9) 派生类。派生类是由基类派生出来的,它继承父类所有或部分方法。除此之外派生类还可以定义自己的方法。

(10) 虚函数。虚函数是在函数前加上关键字 `virtual`。虚函数支持多态性。当虚函数被调用时,适合于当前对象的函数被执行,尽管当前对象的类型直到运行时才能够知道。

以上是 C++ 语言中的一些基本概念,这里只是简单地给以介绍,具体的实现可以查阅相关的面向对象程序设计语言的书籍。

1.3 基本术语

为了更好地理解数据结构这个概念,首先了解一下数据结构中的一些常用名词和术语。

(1) 数据(data)。数据是信息的载体,是描述客观事物的数、字符以及所有能输入到计算机中被计算机程序识别、加工处理的信息的集合。数据不仅仅是通常意义下的整数和实数等。随着计算机的广泛应用,数据的范畴也随之拓广,计算机可以处理的字符串、图像、声音等都可以被称为数据。所以不能只是泛泛的理解数据这个概念。如表 1-1 所示,张风的英语成绩为 92 分,92 就是该同学的成绩数据。

表 1-1 学生成绩表

学号	姓名	语文	数学	英语
S01012	张风	85	69	92
S01022	李强	87	73	74
S02013	王海	92	64	84

(2) 数据项(data item)。数据项是数据的具有独立意义的不可分的最小单位,它是对数据的数据元素属性的描述。数据项也被称为字段或域。

(3) 数据元素(data element)。数据元素是数据的基本单位,是对一个客观实体的数据描述。一个数据元素可以由一个或若干个数据项组成。数据元素也被称为结点或记录。