



普通高等教育“十一五”国家级规划教材

数据结构(C++版)

(第二版)

主 编 李根强

副主编 倪飞舟 钟志水 郭清溥



中国水利水电出版社

www.waterpub.com.cn

普通高等教育“十一五”国家级规划教材

数据结构（C++版）（第二版）

主 编 李根强

副主编 倪飞舟 钟志水 郭清溥



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书为普通高等教育“十一五”国家级规划教材。

全书从软件开发设计的角度出发,按照面向对象程序设计思想,详细介绍线性表、栈和队列、串、多维数组和广义表、树、图等不同的数据结构,以及这些数据结构在计算机中的存储表示和不同存储表示上的算法实现。每个算法都用 C++ 语言进行描述,并全部上机在 Visual C++ 6.0 环境下运行通过。另外介绍计算机中常用的两种运算:查找和排序,详细介绍不同的查找、排序运算的实现及各种算法的效率分析。最后介绍文件的基本概念和文件的组织形式。本书是在 2005 年第一版的基础上,做了一定的修改,增加了相应的功能,涵盖了硕士研究生数据结构考试大纲所规定的考试内容。

本书配套的《数据结构(C++版)(第二版)习题解答及实训指导》一书同时出版,既方便教学,又便于自学。

本书可以作为计算机类或信息类相关专业本科或专科教材及硕士研究生考试参考资料,也可以作为自学数据结构人员的参考资料,还可供从事计算机工程与应用工作的科技人员参考。

本书配有电子教案、源程序及相关文件,读者可以从中国水利水电出版社网站及万水书苑上免费下载,网址为: <http://www.waterpub.com.cn/softdown/>和 <http://www.wsbookshow.com>。

图书在版编目(CIP)数据

数据结构: C++版 / 李根强主编. —2 版. —北京: 中国水利水电出版社, 2009

普通高等教育“十一五”国家级规划教材

ISBN 978-7-5084-6558-6

I. 数… II. 李… 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. TP311.12 TP312

中国版本图书馆 CIP 数据核字(2009)第 081865 号

策划编辑: 雷顺加 责任编辑: 张玉玲 封面设计: 李 佳

书 名	普通高等教育“十一五”国家级规划教材 数据结构(C++版)(第二版)
作 者	主 编 李根强 副主编 倪飞舟 钟志水 郭清溥
出版 发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	184mm×260mm 16 开本 16.75 印张 412 千字
版 次	2005 年 1 月第 1 版 2009 年 6 月第 2 版 2009 年 6 月第 4 次印刷
印 数	11001—15000 册
定 价	26.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

第二版前言

“数据结构”是计算机专业及相关专业的一门重要专业基础课，也是一门必修的核心课程，并且已成为其他理工专业的热门选修课。

在计算机科学的各领域中，都要使用到各种不同的数据结构，如编译系统中要使用栈、散列表、语法树等；操作系统中要使用队列、存储管理表、目录树等；数据库系统中要使用线性表、链表、索引树等；人工智能中要使用广义表、检索树、有向图等；同样在面向对象的程序设计、计算机图形学、软件工程、多媒体技术、计算机辅助设计等领域，都会用到各种不同的数据结构。因此，学好数据结构，对从事计算机技术及相关领域工作的人员来说，是非常重要的，它可以使你掌握各种常用的数据结构及其算法实现，以及每一种算法的时间复杂度分析和空间复杂度分析，知道在什么情况下使用什么样的数据结构最方便，为以后研究和开发大型程序打下基础。

数据结构的主要任务是：讨论现实世界中的各种数据（数字、字符、字符串、声音、图形、图像等）的逻辑结构、在计算机中的各种存储结构（存储表示）以及对各种非数值运算的算法实现；分析各种不同算法的好坏及其在什么地方应用比较合适。通过数据结构课程的学习，使学生具备用所学的数据结构来解决实际问题及评价算法优劣的能力，为以后学习后续计算机专业课程及走上工作岗位从事计算机大型软件开发铺路。

本书是在延续 2005 年第一版编写风格的基础上，根据作者多年教学与研发经验，并考虑到读者使用后的反馈信息，对各章节内容、结构等进行了适当修订、调整、完善和补充，增加了相应的功能，涵盖了硕士研究生数据结构考试大纲所规定的考试内容。

全书内容共分 11 章，第 1 章介绍数据结构与算法等一些基本术语，并对算法描述及算法分析作了简单说明，介绍衡量算法优劣的主要因素：时间复杂度和空间复杂度的求法；第 2 章到第 4 章，介绍线性结构（线性表、栈、队列、串）的逻辑特征、一些常用算法的实现及基本应用；第 5 章到第 7 章，介绍非线性结构（多维数组、广义表、树、二叉树、图）的逻辑特征、在计算机中的存储表示及一些常用算法的实现及基本应用；第 8 章到第 10 章，介绍在计算机中使用非常广泛的两种运算：查找和排序，排序又可以分为内排序和外排序。对一些常用的查找、排序方法进行了详细说明，并给出了实现的算法及时间复杂度和空间复杂度分析；第 11 章介绍外存的文件的几种存储形式及组织方式。各章内容有相对独立的部分，可便于不同院校不同专业按需要组织教学。全书侧重于数据结构的应用，力求讲授内容与具体的计算机应用实例相结合，以便于学生加深对各章内容的理解和掌握。

本书的最大特点是采用面向对象的程序设计语言（C++语言）作为算法的描述语言，所有算法都已经上机调试通过。但是，由于篇幅所限，大部分算法都是以单独的函数形式给出，若读者要运行这些算法，还必须给出一些变量的说明及主函数来调用所给的函数。因此，本书中的算法描述比原来数据结构教材中用类 Pascal 语言或类 C 语言描述算法更直观，学生更容易理解和接受。作者在十几年的数据结构课程教学中，对数据结构中的各种算法进行了认真的研究和分析，在这方面积累了丰富的经验，因此，本书中所选的例题和习题都具有一定的针对性，

都是针对特定的数据结构来进行描述的，方便学生理解和接受，并能为复杂的数据结构算法描述架桥铺路。

本书中所有算法都在 Visual C++ 6.0 环境下运行通过（由于篇幅所限，本书中仅给出了实现某功能算法的函数）。为了方便教学，本书免费为授课教师提供用 PowerPoint 制作的电子教案，教师在使用时可以根据需要进行必要的修改。

本书可以作为高等院校计算机类或信息类相关专业数据结构课程教材，建议讲授课时为 50 至 60 学时，上机实践课时为 20 至 30 学时。各院校、各位教师可根据自己学校的专业特点和学生的实际情况适当增删，目录中前面打*号的可以作为选讲内容。

与本书配套的《数据结构（C++版）（第二版）习题解答及实训指导》一书同时出版，其中提供配套的习题及解答，并配备有数据结构的上机实验指导，可以作为学习指导手册。

本书由李根强任主编，负责全书的统稿、修改、定稿工作，倪飞舟、钟志水、郭清溥任副主编。主要编写人员分工如下：史瑞芳编写了第 1、11 章，倪飞舟编写了第 2、3 章，钟志水编写了第 4、5 章，李根强编写了第 6~8 章，郭清溥编写 9、10 章。参考本书部分编写工作的还有：王刚、汪贵生、姚珺、谢月娥、蔡益红、杜四春、银红霞、袁隽媛、秦志、满淑颖、杜景红等。

由于时间仓促及编者水平有限，书中不妥或错误之处在所难免，恳请专家和广大读者批评指正。

作者

2009 年 2 月

第一版前言

“数据结构”是计算机专业及相关专业的一门重要专业基础课，也是一门必修的核心课程，并且已成为其他理工专业的热门选修课。

在计算机科学的各领域中，都要使用到各种不同的数据结构，如编译系统中要使用栈、散列表、语法树等；操作系统中要使用队列、存储管理表、目录树等；数据库系统中要使用线性表、链表、索引树等；人工智能中要使用广义表、检索树、有向图等；同样在面向对象的程序设计、计算机图形学、软件工程、多媒体技术、计算机辅助设计等领域，都会用到各种不同的数据结构。因此，学好数据结构，对从事计算机技术及相关领域的工作人员来说，是非常重要的，它可以使你掌握各种常用的数据结构及算法实现，以及每一种算法的时间复杂度分析和空间复杂度分析，知道在什么情况下，使用什么样的数据结构最方便，为以后开发大型程序打下基础。

数据结构的主要任务是：讨论现实世界中的各种数据（数字、字符、字符串、声音、图形、图像等）的逻辑结构、在计算机中的各种存储结构（存储表示）以及对各种非数值运算的算法实现；分析各种不同算法的好坏及其在什么地方应用比较合适。通过数据结构课程的学习，使学生具备用所学的数据结构来解决实际问题及评价算法优劣的能力，为以后学习后续计算机专业课程及走上工作岗位从事计算机大型软件开发铺路。

本书内容共分 9 章，第 1 章介绍了数据结构与算法等一些基本术语，并对算法描述及算法分析作了简单说明，介绍了衡量算法优劣的主要因素：时间复杂度和空间复杂度的求法；第 2 章到第 4 章，介绍了线性结构（线性表、栈、队列、串）的逻辑特征，一些常用算法的实现及基本应用；第 5 章到第 7 章，介绍了非线性结构（多维数组、广义表、树、二叉树、图）的逻辑特征，在计算机中的存储表示及一些常用算法实现及基本应用；第 8 章到第 9 章，介绍了在计算机中使用非常广泛的两种运算：查找和排序，对一些常用的查找、排序方法进行了详细说明，并给出了实现的算法及时间复杂度和空间复杂度分析。各章内容有相对独立的部分，可便于不同院校不同专业按需要组织教学。全书侧重于数据结构的应用，力求讲授内容与具体的计算机应用实例相结合，以便于学生加深对各章内容的理解和掌握。

本书的最大特点是采用面向对象的程序设计语言（C++语言）作为算法的描述语言，所有算法都已经上机调试通过。但是，由于篇幅所限，大部分算法都是以单独的函数形式给出，若读者要运行这些算法，还必须给出一些变量的说明及主函数来调用所给的函数。因此，本书中的算法描述比原来数据结构教材中用类 Pascal 语言或类 C 语言描述算法更直观，学生更容易理解和接受。作者在十几年的数据结构课程教学中，对数据结构中的各种算法进行了认真的研究和分析，在这方面积累了丰富的经验，因此，本书中所选的例题和习题都具有一定的针对性，都是针对特定的数据结构来进行描述的，方便学生理解和接受，并能为复杂的数据结构算法描述架桥铺路。

本书中所有算法都在 Visual C++ 6.0 环境下运行通过（由于篇幅所限，本书中仅给出了实现某功能算法的函数）。为了方便教学，本书免费为授课教师提供用 PowerPoint 制作的电子教

案，教师在使用时可以根据需要进行必要的修改。

本书可以作为高等院校计算机类或信息类相关专业数据结构课程教材，建议讲授课时为 50 至 70 学时，上机实践课时为 20 至 30 学时。各院校、各位教师可根据自己学校的专业特点和学生的实际情况，适当增删，目录中前打*号的可以作为选讲内容。

本书也可供从事计算机应用工作的工程与技术人员参考，还可以作为高等院校学生的自学参考书。

与本书配套的还有《数据结构（C++版）习题解答及实习指导》，书中提供配套的习题及解答，并配备有数据结构的上机实验指导，可以作为学习指导手册。

本书由李根强主编，并负责全书的统稿、修改、定稿工作。谢月娥、吴蓉晖、杜四春任副主编。参加本书编写的还有曹翊旺、银红霞、王珺、文斌、刘敏、李红、杜景红、王涛、李辉等，尹海波、王珺、张伟对部分源代码进行了仔细的调试。

由于编者水平有限，书中不妥或错误之处在所难免，恳请专家和广大读者批评指正。

作者

2004 年 10 月

目 录

第二版前言

第一版前言

第1章 绪论	1	本章小结	37
1.1 什么是数据结构	1	习题2	38
1.1.1 数据结构示例	1	第3章 栈和队列	40
1.1.2 基本术语	2	3.1 栈	40
1.1.3 数据结构	3	3.1.1 栈的定义	40
1.2 算法描述	5	3.1.2 栈的运算	41
1.2.1 基本概念	5	3.1.3 栈的抽象数据类型描述	41
1.2.2 算法描述	5	3.1.4 顺序栈	41
1.3 算法分析	6	3.1.5 链栈	45
1.3.1 时间复杂度	7	3.1.6 栈的应用	46
1.3.2 空间复杂度	8	3.2 队列	52
本章小结	8	3.2.1 队列的定义	52
习题1	9	3.2.2 队列的基本运算	52
第2章 线性表	14	3.2.3 队列的抽象数据类型描述	52
2.1 线性表的定义及其运算	14	3.2.4 循环队列	53
2.1.1 线性表的定义	14	3.2.5 链队列	56
2.1.2 线性表的运算	15	3.2.6 队列的应用	58
2.1.3 线性表的抽象数据类型描述	15	本章小结	58
2.2 线性表的顺序存储结构	16	习题3	58
2.2.1 顺序表结构	16	第4章 串	61
2.2.2 顺序表运算	17	4.1 串的定义及运算	61
2.2.3 顺序表存储空间的动态分配	20	4.1.1 基本概念	61
2.3 线性表的链式存储结构	21	4.1.2 串的运算	62
2.3.1 单链表结构	21	4.1.3 串的抽象数据类型描述	62
2.3.2 单链表运算	22	4.2 串的存储结构	62
2.3.3 循环链表结构	27	4.2.1 顺序存储	62
2.3.4 双向链表结构	29	4.2.2 链式存储	64
2.4 一元多项式的表示及相加	31	4.2.3 索引存储	65
2.4.1 一元多项式的表示	31	4.3 串运算的实现	65
2.4.2 一元多项式的相加	32	4.3.1 串插入	65
2.5 顺序表与链表的比较	34	4.3.2 串删除	67
2.6 算法应用举例	35	4.3.3 子串定位	68

4.4 串操作应用举例	70	6.3.4 遍历算法应用举例	113
4.4.1 文本编辑	70	6.4 线索二叉树	117
4.4.2 建立词索引表	72	6.4.1 线索的概念	117
本章小结	72	6.4.2 线索的描述	118
习题 4	72	6.4.3 线索的算法实现	120
第 5 章 多维数组和广义表	74	6.4.4 线索二叉树上的运算	120
5.1 多维数组	74	6.5 树和森林	123
5.1.1 多维数组的概念	74	6.5.1 树的存储结构	123
5.1.2 多维数组在计算机内的存储	75	6.5.2 树、森林和二叉树的转换	125
5.2 多维数组的存储结构	75	6.5.3 树和森林的遍历	127
5.2.1 行优先顺序	75	6.6 回溯法与树的遍历	128
5.2.2 列优先顺序	76	6.7 哈夫曼树	129
5.3 特殊矩阵及其压缩存储	76	6.7.1 基本术语	129
5.3.1 特殊矩阵	76	6.7.2 哈夫曼树简介	129
5.3.2 压缩存储	77	6.7.3 哈夫曼树的应用	133
5.4 稀疏矩阵	79	本章小结	133
5.4.1 稀疏矩阵的存储	80	习题 6	134
5.4.2 稀疏矩阵的运算	82	第 7 章 图	137
5.5 广义表	91	7.1 图的基本概念	137
5.5.1 基本概念	91	7.1.1 图的定义	137
5.5.2 存储结构	92	7.1.2 图的基本术语	137
5.5.3 基本运算	94	7.2 图的存储结构	140
本章小结	96	7.2.1 邻接矩阵	140
习题 5	96	7.2.2 邻接表	143
第 6 章 树和二叉树	99	7.2.3 邻接多重表	147
6.1 树的基本概念	99	7.3 图的遍历	148
6.1.1 树的定义	99	7.3.1 深度优先搜索遍历	148
6.1.2 基本术语	100	7.3.2 广度优先搜索遍历	153
6.1.3 树的表示	101	7.4 生成树和最小生成树	157
6.1.4 树的性质	102	7.4.1 基本概念	157
6.2 二叉树	102	7.4.2 普里姆 (prim) 算法	158
6.2.1 二叉树的定义	102	7.4.3 克鲁斯卡尔 (kruskal) 算法	161
6.2.2 二叉树的性质	103	7.5 最短路径	164
6.2.3 二叉树的存储结构	105	7.5.1 单源点最短路径	164
6.2.4 二叉树的抽象数据类型	108	7.5.2 所有顶点对之间的最短路径	167
6.3 遍历二叉树	108	7.6 有向无环图及其应用	170
6.3.1 前根遍历	109	7.6.1 拓扑排序	171
6.3.2 中根遍历	110	7.6.2 关键路径	175
6.3.3 后根遍历	111	本章小结	179

习题 7	180	9.4.2 树型选择排序	226
第 8 章 查找	183	9.4.3 堆排序	227
8.1 查找的基本概念	183	9.5 归并排序	232
8.2 线性表的查找	184	9.5.1 二路归并排序	232
8.2.1 顺序查找	184	9.5.2 多路归并排序	234
8.2.2 二分查找	185	9.6 分配排序	234
8.2.3 索引查找	188	9.6.1 多关键字排序	234
8.2.4 分块查找	192	9.6.2 链式基数排序	235
8.3 树表查找	193	9.7 各种内排序方法的比较和选择	238
8.3.1 二叉排序树查找	193	9.7.1 各种内排序方法的比较	238
8.3.2 平衡二叉树查找	198	9.7.2 各种内排序方法的选择	238
8.3.3 B 树及 B 树上的查找	202	本章小结	239
8.3.4 键树	204	习题 9	239
8.4 散列查找	204	第 10 章 外排序	241
8.4.1 基本概念	204	10.1 外排序的基本概念	241
8.4.2 散列函数的构造	205	10.2 多路平衡归并的实现	242
8.4.3 解决冲突的方法	207	10.2.1 初始归并段的生成	242
8.4.4 散列查找算法的实现	210	10.2.2 多路平衡归并的实现	244
8.4.5 散列查找的性能分析	212	本章小结	249
本章小结	214	习题 10	249
习题 8	215	第 11 章 文件	250
第 9 章 内排序	216	11.1 文件的基本概念	250
9.1 基本概念	216	11.2 顺序文件	250
9.1.1 排序介绍	216	11.3 索引文件	251
9.1.2 基本概念	217	11.4 ISAM 文件和 VSAM 文件	252
9.2 插入排序	218	11.4.1 ISAM 文件	252
9.2.1 直接插入排序	218	11.4.2 VSAM 文件	253
9.2.2 二分插入排序	219	11.5 散列文件	253
9.2.3 希尔排序	220	11.6 多关键字文件	254
9.3 交换排序	221	11.6.1 多重表文件	255
9.3.1 冒泡排序	221	11.6.2 倒排文件	256
9.3.2 快速排序	223	本章小结	256
9.4 选择排序	225	习题 11	257
9.4.1 直接选择排序	225	参考文献	258

第 1 章 绪论

本章导读

本章主要介绍数据结构课程中的一些常用术语以及集合、线性结构、树型结构和图型结构等常用数据结构的表示，用 C++ 实现算法描述的一般规则，算法的时间复杂度和空间复杂度分析与评价。

本章要点

- 数据结构中的常用基本术语
- 集合、线性结构、树型结构和图型结构等每一种常用数据结构的逻辑特点
- 抽象数据类型的定义、使用，算法的定义、特性及用 C++ 描述算法的规则
- 评价算法优劣的规则，算法的时间复杂度、空间复杂度的定义及数量级的表示
- 复习 C++ 中的有关语法规则，以便满足在数据结构课程中进行算法描述的需要

1.1 什么是数据结构

1.1.1 数据结构示例

为了使大家对数据结构有个感性认识，先举出几个例子来说明数据结构。

【例 1-1】给出一张学生数据表，如表 1-1 所示。

表 1-1 学生数据表

学号	姓名	性别	籍贯	电话	通讯地址
01	张三	男	长沙	8639000	麓山南路 327 号
02	李四	男	北京	23456789	学院路 435 号
03	王五	女	广州	30472589	天河路 478 号
04	赵六	男	上海	41237568	南京路 1563 号
05	钱七	女	南京	5013472	南京大学
06	刘八	女	武汉	61543726	武汉大学
07	朱九	男	昆明	4089651	云南大学
08	孙十	女	杭州	6154372	西湖路 635 号

在学生表中，一行为一个学生信息，代表一个学生数据，一列为一个属性，整个二维表格形成学生数据的一个线性序列，每个学生排列的位置有先后次序，它们之间形成一种线性关系，是一种典型的数据结构（线性结构），我们将它称为线性表。

【例 1-2】描述一个磁盘的目录及文件结构，包含一个根目录、若干个一级子目录（文件夹），每个一级子目录中又包含若干个二级子目录（子文件夹），如图 1-1 所示。

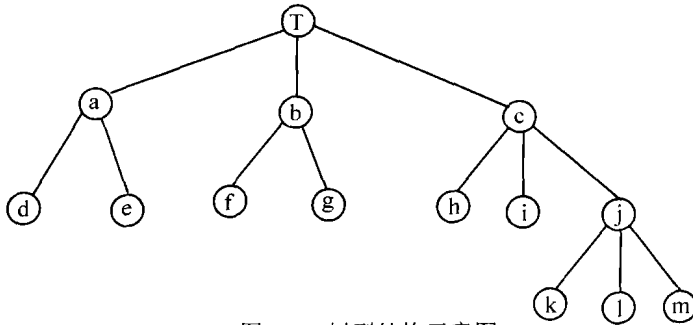


图 1-1 树型结构示意图

在此种结构中，数据之间呈现一对多的非线性关系，也是常用的一种数据结构（非线性结构），我们将它称为树型结构。

【例 1-3】描述一个大学的校园网，圆圈代表站点，边表示网线，如图 1-2 所示。

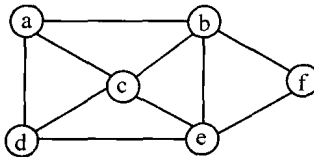


图 1-2 图型结构示意图

在此种结构中，数据之间呈现多对多的非线性关系，这也是常用的一种数据结构（非线性结构），我们将它称为图型结构。

综合上述 3 个例题，我们将对数据结构及其一些基本术语作进一步说明。

1.1.2 基本术语

1. 数据 (Data)

数据是指能够输入到计算机中，并被计算机识别和处理的符号的集合。

例如，数字、字母、汉字、图形、图像、声音都称为数据。

2. 数据元素 (Data Element)

数据元素是组成数据的基本单位。

数据元素是一个数据整体中相对独立的单位，但它还可以分割成若干个具有不同属性的项（字段），故不是组成数据的最小单位。

3. 数据对象 (Data Object)

数据对象是性质相同的数据元素组成的集合，是数据的一个子集。

例如，整数数据对象的集合可表示为 $N=\{0, \pm 1, \pm 2, \dots\}$ ，大写字母字符数据对象的集合可表示为 $C=\{'A', 'B', \dots, 'Z'\}$ 。

4. 数据类型 (Data Type)

数据是一组性质相同的值的集合以及定义于这个值集合上的一组操作的总称。

例如，高级语言中用到的整数数据类型，是指由-32768~32767 中的整数数值构成的集合及一组操作（加、减、乘、除、乘方等）的总称。

5. 抽象数据类型 (Abstract Data Types)

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，抽象数据类型由基本数据类型组成，并包括一组相关的操作。抽象数据类型有些类似于 C/C++ 语言中的 struct 类型 and Pascal 语言中的 record 类型，但它增加了相关的操作。

【例 1-4】给出自然数 (Natural Number) 的抽象数据类型定义。

ADT Natural Number is

Data:

一个整数的有序子集合，它开始于 0，终止于机器能表示的最大整数 (MAXINT)。

Operation:

对于所有 $x, y \in \text{Natural Number}$ ，定义如下操作：

add(x,y)	求 $X+Y$
sub(x,y)	求 $X-Y$
mul(x,y)	求 $X \times Y$
div(x,y)	求 $X \div Y$
equal(x,y)	判断 X、Y 是否相等

END

在本书中，描述一种抽象数据类型将采用如下书写格式：

ADT <抽象数据类型名> is

Data: <数据描述>

Operation:<操作声明>

END

1.1.3 数据结构

1. 数据结构 (Data Structure)

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所组成的集合。具体来说，数据结构包含 3 个方面的内容，即数据的逻辑结构、数据的存储结构和对数据所施加的运算。这 3 个方面的关系为：

(1) 数据的逻辑结构独立于计算机，是数据本身所固有的。

(2) 存储结构是逻辑结构在计算机存储器中的映像，必须依赖于计算机。

(3) 运算是指所施加的一组操作的总称。运算的定义直接依赖于逻辑结构，但运算的实现必依赖于存储结构。

比如，例 1-1 提到的学生数据表，除了有 8 个学生的数据外，还存在着一对一的线性关系；例 1-2 提到的磁盘目录及文件结构，除包含文件数据外，还存在着目录之间一对多的非线性关系；例 1-3 提到的大学校园网，除包含站点数据外，还存在着站点间的多对多的非线性关系。

2. 从逻辑结构划分数据结构

数据结构从逻辑结构上划分为：

(1) 线性结构。

元素之间为一对一的线性关系, 第一个元素无直接前驱, 最后一个元素无直接后继, 其余元素都有一个直接前驱和直接后继。

(2) 非线性结构。

元素之间为一对多或多对多的非线性关系, 每个元素有多个直接前驱或多个直接后继。

(3) 集合结构。

元素之间无任何关系, 元素的排列无任何顺序。

3. 从存储结构划分数据结构

数据结构从存储结构上划分为:

(1) 顺序存储 (向量存储)。

所有元素存放在一片连续的存储单元中, 逻辑上相邻的元素存放到计算机内存中仍然相邻。

(2) 链式存储。

所有元素存放在可以不连续的存储单元中, 但元素之间的关系可以通过地址确定, 逻辑上相邻的元素存放到计算机内存后不一定是相邻的。

(3) 索引存储。

使用该方法存放元素的同时, 还建立附加的索引表, 索引表中的每一项称为索引项, 索引项的一般形式是: (关键字, 地址), 其中的关键字是能唯一标识一个结点的那些数据项。

(4) 散列存储。

通过构造散列函数, 用函数的值来确定元素存放的地址。

4. 数据结构的抽象描述

数据结构可用二元组 $D=(K,R)$ 的形式来描述。其中, $K=\{a_1, a_2, \dots, a_n\}$ 为元素的集合, $R=\{r_1, r_2, \dots, r_m\}$ 为关系的集合。

【例 1-5】设有一个线性表 $(a_1, a_2, a_3, a_4, a_5)$, 它的抽象描述可表示为 $D=(K,R)$, 其中 $K=\{a_1, a_2, a_3, a_4, a_5\}$, $R=\{<a_1, a_2>, <a_2, a_3>, <a_3, a_4>, <a_4, a_5>\}$, 则它的逻辑结构的图形描述如图 1-3 所示。

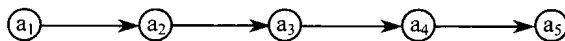


图 1-3 线性表结构抽象描述示意图

【例 1-6】设一个数据结构的抽象描述为 $D=(K,R)$, 其中 $K=\{a,b,c,d,e,f,g,h\}$, $R=\{<a,b>, <a,c>, <a,d>, <b,e>, <c,f>, <c,g>, <d,h>\}$, 则它的逻辑结构的图形描述如图 1-4 所示。

【例 1-7】设一个数据结构的抽象描述为 $D=(K,R)$, 其中 $K=\{1,2,3,4\}$, $R=\{(1,2), (1,3), (1,4), (2,3), (2,4), (3,4)\}$, 则它的逻辑结构的图形描述如图 1-5 所示。

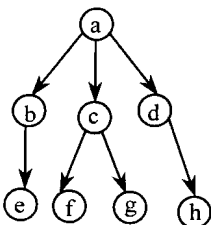


图 1-4 树型结构抽象描述示意图

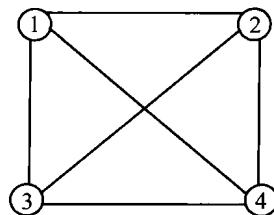


图 1-5 图型结构抽象描述示意图

1.2 算法描述

1.2.1 基本概念

1. 算法 (Algorithm)

通俗地讲, 算法就是一种解题的方法。更严格地说, 算法是由若干条指令组成的有穷序列, 它必须满足下述条件 (也称为算法的五大特性):

- (1) 输入: 具有 0 个或多个输入的外界量 (算法开始前的初始量)。
- (2) 输出: 至少产生一个输出, 它们是算法执行完后的结果。
- (3) 有穷性: 每条指令的执行次数必须是有限的。
- (4) 确定性: 每条指令的含义都必须明确、无二义性。
- (5) 可行性: 每条指令的执行时间都是有限的。

2. 算法和程序的关系

算法的含义与程序十分相似, 但二者是有区别的。一个程序不一定满足有穷性 (死循环), 另外, 程序中的指令必须是机器可执行的, 而算法中的指令则无此限制。一个算法若用计算机语言来书写, 则它就可以是一个程序。

1.2.2 算法描述

1. 用流程图描述算法

一个算法可以用流程图的方式来描述, 输入输出、判断、处理分别用不同的框图表示, 用箭头表示流程的流向。这是一种描述算法的较好方法, 目前在一些高级语言程序设计中仍然采用。

2. 用自然语言描述算法

用我们日常生活中的自然语言 (可以是中文形式, 也可以是英文形式) 也可以描述算法。例如, 某同志某天做的工作可以描述为一个算法形式, 如下:

若今天我有空并且天不下雨, 则我上街购物, 否则我呆在家里看书。

3. 用其他方式描述算法

还可以用数学语言或约定的符号语言来描述算法。

4. 用 C++ 描述算法

在本教材中, 将采用 C++ 或类 C++ 来描述算法。用 C++ 描述算法遵循如下规则:

- (1) 所有算法的描述都用 C++ 中的函数形式。

函数类型 函数名(形参及类型说明)

```
{ 函数语句部分
```

```
    return(表达式值);
```

```
}
```

- (2) 函数中的形式参数有两种传值方式。

若为一般变量名, 则为单向传值参数; 若在变量前面增加 & 符号, 则为双向传地址参数。例如有一个函数为 `void swap(&i, &j, k)`, 则 `i`、`j` 为双向传地址参数, `k` 为单向传值参数。

(3) 函数的说明部分与函数的实现部分分离。

在 C++ 中, 用函数来描述算法时, 为使之与面向对象的程序设计相匹配, 一般将函数的说明部分与函数的实现部分分离开来。

(4) 输入函数。

C++ 中的输入函数调用为: `cin>>变量名`。

(5) 输出函数。

C++ 中的输出函数调用为: `cout<<变量名`。

(6) C++ 中的类。

C++ 对于面向对象程序设计的支持, 核心部分就是类的定义。类的定义体现了抽象数据类型的思想, 可以支持说明与实现的分离, 将抽象数据类型的实现封装在类的内部, 使之达到信息隐蔽的目的。

(7) `public` 说明。

对于 C++ 的类, 类成员可以用 `public` 声明, 表示这些东西是公有的, 可以在此类及其他类中使用。

(8) `private` 说明。

对于 C++ 类, 类成员可以用 `private` 声明, 表示这些东西是私有的, 只能在本类中使用。

(9) `protected` 说明。

对于 C++ 类, 类成员可以用 `protected` 声明, 表示这部分是受保护的, 只能在本类及它的所有派生类中使用。

(10) C++ 的作用域。

在 C++ 中, 每个变量都有一个作用域。在函数内声明的变量, 仅能在该函数内部有效; 在类中声明的变量, 可以在该类内部有效; 在整个程序中都能使用的变量, 称为全局变量, 否则称为局部变量。若一个全局变量与一个局部变量同名, 则该范围内全局变量不起作用。若要使之起作用, 可以使用域操作符 `::` 来实现。

(11) 函数名重载。

C++ 中允许多个函数取相同的函数名, 但形式参数和返回类型可以不同。

(12) 友元函数。

在 C++ 的类声明中, 可以使用保留字 `friend` 定义友元函数, 使用友元函数可以在一个类中访问另一个类中的私有成员 (`private`) 和被保护成员 (`protected`)。

(13) 内联 (`inline`) 函数。

在一个函数定义前加上 `inline` 前缀就成为内联函数。使用内联函数可以减少函数调用和参数传递的系统开销。

1.3 算法分析

求解同一个问题, 可以有许多不同的算法, 那么怎样来衡量这些算法的优劣呢? 首要的条件是选用的算法必须是正确的, 其次考虑如下 3 点:

(1) 执行算法所耗费的时间。

(2) 执行算法所占用的内存开销 (主要考虑占用的辅助存储空间)。

(3) 算法应易于理解、易于编码、易于调试等。

1.3.1 时间复杂度

1. 时间频度

一个算法执行所耗费的时间，从理论上是不能算出来的，必须上机运行测试才能知道。但我们不可能也没有必要对每个算法都上机测试（因为，计算机的运行速度与 CPU 等因素有关。同一算法在不同的计算机上运行的时间是不同的），只需知道在相同的条件下，哪个算法花费的时间多，哪个算法花费的时间少就可以了。并且一个算法花费的时间与算法中语句的执行次数成正比，哪个算法中语句执行次数多，它花费的时间就多。一个算法中的语句执行次数称为语句频度或时间频度，记为 $T(n)$ 。

【例 1-8】求下列算法段的语句频度。

```
for(i=1; i<=n; i++)
  for(j=1; j<=i; j++)
    x=x+1;
```

分析：该算法为一个二重循环，执行次数为内、外循环次数相乘，但内循环次数不固定，与外循环有关，因此，时间频度 $T(n)=1+2+3+\dots+n=\frac{n(n+1)}{2}$ 。

2. 时间复杂度

在刚才提到的时间频度中， n 称为问题的规模，当 n 不断变化时，时间频度 $T(n)$ 也会不断变化。但有时我们想知道它变化时呈现什么规律，为此，引入时间复杂度概念。

设 $T(n)$ 的一个辅助函数为 $g(n)$ ，定义为当 n 大于等于某一足够大的正整数 n_0 时，存在两个正的常数 A 和 B （其中 $A \leq B$ ），使得 $A \leq \frac{T(n)}{g(n)} \leq B$ 均成立，或有 $\lim_{n \rightarrow +\infty} \frac{T(n)}{g(n)} = A$ （其中 A 为常数），则称 $g(n)$ 是 $T(n)$ 的同数量级函数。把 $T(n)$ 表示成数量级的形式为： $T(n)=O(g(n))$ ，其中大写字母 O 为英文 Order（即数量级）一词的第一个字母。

例如，若 $T(n)=\frac{n(n+1)}{2}$ ，则有 $\lim_{n \rightarrow +\infty} \frac{T(n)}{n^2} = \frac{1}{2}$ ，故它的时间复杂度为 $O(n^2)$ ，即 $T(n)$ 与 n^2 数量级相同。

【例 1-9】分析下列算法段的时间频度及时间复杂度。

```
for (i=1; i<=n; i++)
  for (j=1; j<=i; j++)
    for (k=1; k<=j; k++)
      x=i+j-k;
```

分析算法规律可知时间频度 $T(n)=1+(1+2)+(1+2+3)+\dots+(1+2+3+\dots+n)$

$$\begin{aligned} &= \sum_{k=1}^n (1+2+3+\dots+k) \\ &= \sum_{k=1}^n \frac{k(k+1)}{2} \\ &= \sum_{k=1}^n \frac{k^2}{2} + \sum_{k=1}^n \frac{k}{2} \end{aligned}$$