

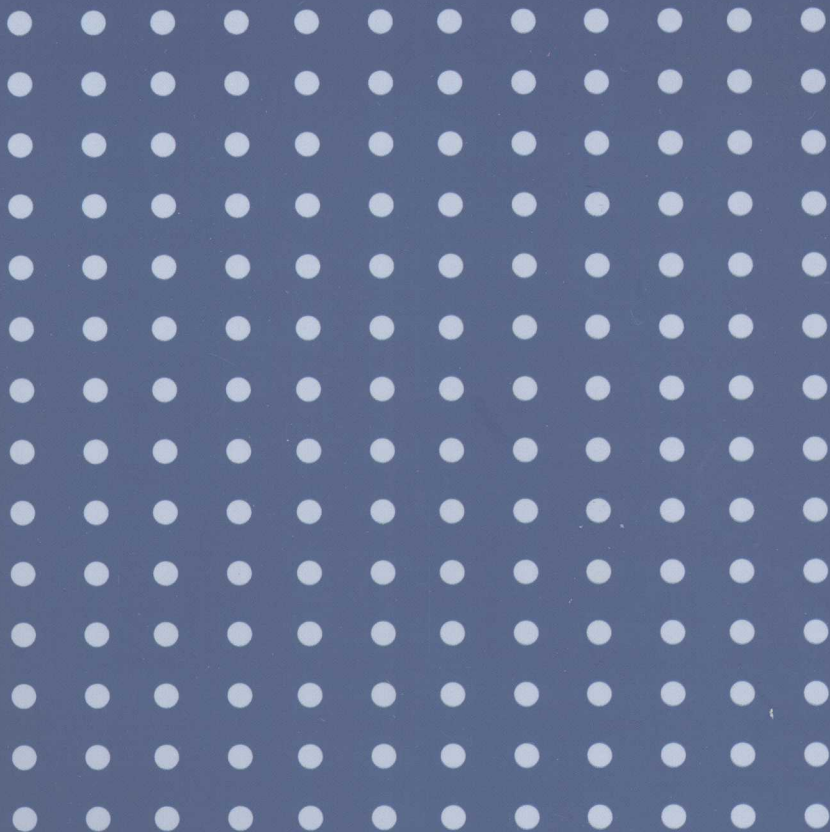


普通高等教育“十一五”国家级规划教材

重点大学计算机专业系列教材

数据结构教程(第3版)

李春葆 尹为民 李蓉蓉 蒋晶珏 喻丹丹 安杨 编著



清华大学出版社



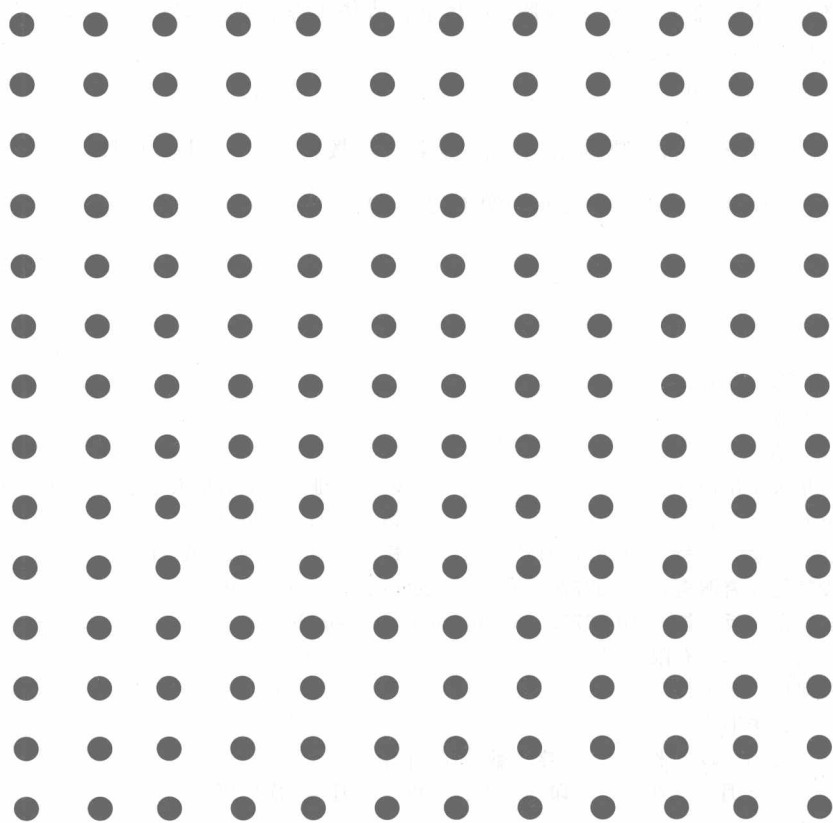


普通高等教育“十一五”国家级规划教材

重点大学计算机专业系列教材

数据结构教程(第3版)

李春葆 尹为民 李蓉蓉 蒋晶珏 喻丹丹 安杨 编著



清华大学出版社

北京

内 容 简 介

本书在前两版的基础上,针对教育部新的考研大纲和大量读者来信提出的要求进行了修订。本书内容包括绪论、线性表、栈和队列、串、数组和广义表、递归、树形结构、图、查找、内排序、外排序和文件,还给出了3个综合实验题、实验报告格式、引用型参数的说明、顺序表和顺序栈以及顺序队列使用指针引用型参数的说明、书中部分算法清单、2009年全国计算机专业数据结构考研大纲。

本书适合高等院校计算机及相关专业本科生及研究生使用。

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。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

数据结构教程/李春葆等编著. —3版. —北京:清华大学出版社,2009.3
(重点大学计算机专业系列教材)

ISBN 978-7-302-19379-1

I. 数… II. 李… 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(2009)第 012401 号

责任编辑:丁 岭 李玮琪

责任校对:焦丽丽

责任印制:何 芊

出版发行:清华大学出版社

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

社 总 机:010-62770175

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮 购:010-62786544

印 刷 者:北京季蜂印刷有限公司

装 订 者:三河市溧源装订厂

经 销:全国新华书店

开 本:185×260 印 张:23.5 字 数:572 千字

版 次:2009年3月第3版 印 次:2009年3月第1次印刷

印 数:1~3000

定 价:36.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:031985-01

出版说明

随着国家信息化步伐的加快和高等教育规模的扩大,社会对计算机专业人才的需求不仅体现在数量的增加上,而且体现在质量要求的提高上,培养具有研究和实践能力的高层次的计算机专业人才已成为许多重点大学计算机专业教育的主要目标。目前,我国共有16个国家重点学科、20个博士点一级学科、28个博士点二级学科集中在教育部部属重点大学,这些高校在计算机教学和科研方面具有一定优势,并且大多以国际著名大学计算机教育为参照系,具有系统完善的教学课程体系、教学实验体系、教学质量保证体系和人才培养评估体系等综合体系,形成了培养一流人才的教学和科研环境。

重点大学计算机学科的教学与科研氛围是培养一流计算机人才的基础,其中专业教材的使用和建设则是这种氛围的重要组成部分,一批具有学科方向特色优势的计算机专业教材作为各重点大学的重点建设项目成果得到肯定。为了展示和发扬各重点大学在计算机专业教育上的优势,特别是专业教材建设上的优势,同时配合各重点大学的计算机学科建设和专业课程教学需要,在教育部相关教学指导委员会专家的建议和各重点大学的大力支持下,清华大学出版社规划并出版本系列教材。本系列教材的建设旨在“汇聚学科精英、引领学科建设、培育专业英才”,同时以教材示范各重点大学的优秀教学理念、教学方法、教学手段和教学内容等。

本系列教材在规划过程中体现了如下一些基本组织原则和特点。

1. 面向学科发展的前沿,适应当前社会对计算机专业高级人才的培养需求。教材内容以基本理论为基础,反映基本理论和原理的综合应用,重视实践和应用环节。

2. 反映教学需要,促进教学发展。教材要能适应多样化的教学需要,正确把握教学内容和课程体系的改革方向。在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

3. 实施精品战略,突出重点,保证质量。规划教材建设的重点依然是专业基础课和专业主干课;特别注意选择并安排了一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现重点大学

计算机专业教学内容和课程体系改革成果的教材。

4. 主张一纲多本,合理配套。专业基础课和专业主干课教材要配套,同一门课程可以有多本具有不同内容特点的教材。处理好教材统一性与多样化的关系;基本教材与辅助教材以及教学参考书的关系;文字教材与软件教材的关系,实现教材系列资源配套。

5. 依靠专家,择优落实。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

教材编委会

前言

数据结构是计算机学科的必修课程,涵盖了计算机学科的算法设计、数值分析、操作系统和编译原理等课程涉及的大部分相关算法的实现。学好该课程不仅对这些后续课程的学习有很大帮助,而且在实际中有广泛的用途。

计算机是进行数据处理的工具,数据结构主要研究数据的各种组织形式以及建立在这些结构之上的各种运算算法的实现,它不仅为用计算机语言进行程序设计提供了方法性的理论指导,还在更高的层次上总结了程序设计的常用方法和常用技巧。

本教程是作者针对数据结构课程概念多、算法灵活和抽象性强等特点,在总结长期教学经验的基础上编写的。全书分为 13 章和 6 个附录,第 1 章为绪论,介绍数据结构的基本概念,特别强调算法分析的方法;第 2 章为线性表,介绍线性表的两种存储结构——顺序表和链表与基本运算算法的实现过程;第 3 章为栈和队列,介绍这两种特殊线性结构的概念与应用;第 4 章为串,介绍串的概念与模式匹配算法;第 5 章为数组和广义表,介绍数组、稀疏矩阵和广义表的概念与相关运算算法的实现过程;第 6 章为递归,较深入地讨论了计算机学科中递归算法设计方法;第 7 章为树形结构,介绍树和二叉树的概念与各种运算算法的实现过程,其中特别突出二叉树的各种递归算法方法;第 8 章为图,介绍图的概念和图的各种运算算法的实现过程;第 9 章为查找,介绍各种查找算法的实现过程;第 10 章为内排序,介绍各种内排序算法的实现过程;第 11 章为外排序,介绍各种外排序算法的实现过程;第 12 章为文件,介绍各类文件的组织结构;第 13 章为采用面向对象方法描述算法,介绍面向对象的概念和用 C++ 描述数据结构算法的方法。附录 A 给出了 3 个综合实验题;附录 B 给出了实验报告格式;附录 C 是引用型参数的说明;附录 D 是顺序表、顺序栈和顺序队列的算法中使用指针引用型参数的说明;附录 E 给出了书中部分算法清单;附录 G 为教育部颁发的 2009 年全国计算机专业数据结构部分考研大纲。

数据结构是一门应用性非常强的课程,学生在掌握各种数据结构特别是存储结构的基础上,一定要尽可能多地上机实习,通过较多的实验把难以理解的抽象概念转化为实实在在的计算机能够正确运行的程序,这样才能将所

学知识和实际应用结合起来,吸取算法的设计思想和精髓,提高运用这些知识解决实际问题的能力。因此,本教程突出上机实习内容,除最后一章外其余各章都给出大量的上机实验题(属验证设计型实验),供教师和学生选用,附录 A 还给出 3 个综合性较强的实验题(属综合设计型实验),目的是全面考查学生综合运用数据结构知识的能力,一般在本课程学习末期或者在专门的数据结构集中实习课(通常为 36 课时)中向学生布置。

为了便于学习和上机实验,我们还编写了与本教程配套的《数据结构教程学习指导》和《数据结构教程上机实验指导》两本书,构成一个完整的教学系列。本系列中所有程序均在 Visual C++ 6.0 环境下调试通过。

本教程和相关学习指导的编写得到武汉大学教务部“数据结构综合教学改革”教学项目的支持,是本群组许多教师多年来在数据结构课程教学研究和教学改革中的经验与成果的结晶,特别得到黄传河、黄竟伟、黄水松、薛超英等多位教授和博导的大力帮助。作者在此表示衷心感谢。

本书在前两版的基础上,针对教育部颁发的新的考研大纲和大量读者来信提出的要求进行了修订,例如,广义表在新考纲中没有出现,将其合并到数组部分,同时对全书中多个算法进行了优化。另外,编者在 VC++ 6.0 环境中实现了书中各数据结构的基本运算算法和算法设计例题,这些源程序和本书的课件可以免费从 <http://www.tup.com.cn> 网站下载。

由于水平所限,可能存在错误和不足之处,敬请读者批评指正,特别希望使用本书的教师与作者探讨,共同提高我国计算机专业数据结构课程的教学水平。作者的联系信箱为 licb1964@126.com。

编者

2008年10月

目录

第 1 章 绪论	1
1.1 什么是数据结构	1
1.1.1 数据结构的定义	1
1.1.2 逻辑结构类型	5
1.1.3 存储结构类型	7
1.1.4 数据结构和数据类型	8
1.2 算法及其描述	11
1.2.1 什么是算法	11
1.2.2 算法描述	12
1.3 算法分析	14
1.3.1 算法设计的目标	14
1.3.2 算法效率分析	14
1.3.3 算法存储空间分析	17
1.4 数据结构+算法=程序	18
本章小结	24
练习题 1	24
上机实验题 1	26
第 2 章 线性表	27
2.1 线性表及其逻辑结构	27
2.1.1 线性表的定义	27
2.1.2 线性表的抽象数据类型描述	28
2.2 线性表的顺序存储结构	29
2.2.1 线性表的顺序存储结构——顺序表	29
2.2.2 顺序表基本运算的实现	30
2.3 线性表的链式存储结构	36
2.3.1 线性表的链式存储结构——链表	36

2.3.2	单链表基本运算的实现	38
2.3.3	双链表	44
2.3.4	循环链表	48
2.4	线性表的应用	49
2.5	有序表	53
	本章小结	56
	练习题 2	56
	上机实验题 2	56
第 3 章	栈和队列	59
3.1	栈	59
3.1.1	栈的定义	59
3.1.2	栈的顺序存储结构及其基本运算实现	61
3.1.3	栈的链式存储结构及其基本运算的实现	63
3.1.4	栈的应用举例	66
3.2	队列	76
3.2.1	队列的定义	76
3.2.2	队列的顺序存储结构及其基本运算的实现	77
3.2.3	队列的链式存储结构及其基本运算的实现	80
3.2.4	队列的应用举例	83
	本章小结	87
	练习题 3	87
	上机实验题 3	88
第 4 章	串	91
4.1	串的基本概念	91
4.2	串的存储结构	92
4.2.1	串的顺序存储结构——顺序串	92
4.2.2	串的链式存储结构——链串	97
4.3	串的模式匹配	103
4.3.1	Brute-Force 算法	103
4.3.2	KMP 算法	105
	本章小结	111
	练习题 4	111
	上机实验题 4	111
第 5 章	数组和广义表	114
5.1	数组	114
5.1.1	数组的基本概念	114

5.1.2	数组的存储结构	115
5.1.3	特殊矩阵的压缩存储	117
5.2	稀疏矩阵	119
5.2.1	稀疏矩阵的三元组表示	119
5.2.2	稀疏矩阵的十字链表表示	123
5.3	广义表	126
5.3.1	广义表的定义	126
5.3.2	广义表的存储结构	128
5.3.3	广义表的运算	129
	本章小结	132
	练习题 5	132
	上机实验题 5	133
第 6 章	递归	135
6.1	什么是递归	135
6.1.1	递归的定义	135
6.1.2	何时使用递归	136
6.1.3	递归模型	137
6.1.4	递归与数学归纳法	138
6.2	递归调用的实现原理	139
6.3	递归算法的设计	141
6.3.1	递归算法设计的步骤	141
6.3.2	递归数据结构的递归算法设计	142
6.3.3	递归求解方法的递归算法设计	143
	本章小结	145
	练习题 6	145
	上机实验题 6	146
第 7 章	树和二叉树	147
7.1	树的基本概念	147
7.1.1	树的定义	147
7.1.2	树的逻辑表示方法	148
7.1.3	树的基本术语	149
7.1.4	树的性质	150
7.1.5	树的基本运算	151
7.1.6	树的存储结构	152
7.2	二叉树的概念和性质	155
7.2.1	二叉树的概念	155
7.2.2	二叉树的性质	156

7.2.3	二叉树与树、森林之间的转换	157
7.3	二叉树存储结构	159
7.3.1	二叉树的顺序存储结构	159
7.3.2	二叉树的链式存储结构	161
7.4	二叉树的基本运算及其实现	161
7.4.1	二叉树的基本运算概述	161
7.4.2	二叉树的基本运算算法实现	162
7.5	二叉树的遍历	165
7.5.1	二叉树遍历的概念	165
7.5.2	二叉树遍历递归算法	166
7.5.3	二叉树遍历非递归算法	169
7.5.4	层次遍历算法	174
7.6	二叉树的构造	176
7.7	线索二叉树	181
7.7.1	线索二叉树的概念	181
7.7.2	线索化二叉树	183
7.7.3	遍历线索化二叉树	184
7.8	哈夫曼树	185
7.8.1	哈夫曼树概述	185
7.8.2	哈夫曼树的构造算法	186
7.8.3	哈夫曼编码	187
7.9	用并查集求解等价问题	189
7.9.1	什么叫并查集	189
7.9.2	并查集的算法实现	191
	本章小结	193
	练习题 7	193
	上机实验题 7	194
第 8 章	图	196
8.1	图的基本概念	196
8.1.1	图的定义	196
8.1.2	图的基本术语	197
8.2	图的存储结构	199
8.2.1	邻接矩阵存储方法	200
8.2.2	邻接表存储方法	201
8.3	图的遍历	204
8.3.1	图的遍历的概念	204
8.3.2	深度优先搜索遍历	204
8.3.3	广度优先搜索遍历	205

8.3.4	非连通图的遍历	206
8.3.5	图遍历算法的应用	207
8.4	生成树和最小生成树	210
8.4.1	生成树的概念	210
8.4.2	无向图的连通分量和生成树	211
8.4.3	普里姆算法	212
8.4.4	克鲁斯卡尔算法	214
8.5	最短路径	218
8.5.1	路径的概念	218
8.5.2	从一个顶点到其余各顶点的最短路径	218
8.5.3	每对顶点之间的最短路径	223
8.6	拓扑排序	227
8.7	AOE网与关键路径	229
	本章小结	233
	练习题 8	233
	上机实验题 8	233
第 9 章	查找	236
9.1	查找的基本概念	236
9.2	线性表的查找	237
9.2.1	顺序查找	237
9.2.2	二分查找	238
9.2.3	索引存储结构和分块查找	240
9.3	树表的查找	243
9.3.1	二叉排序树	243
9.3.2	平衡二叉树	250
9.3.3	B-树	256
9.3.4	B+树	262
9.4	哈希表查找	263
9.4.1	哈希表的基本概念	263
9.4.2	哈希函数构造方法	264
9.4.3	哈希冲突解决方法	265
9.4.4	哈希表上的运算	268
	本章小结	271
	练习题 9	271
	上机实验题 9	272
第 10 章	内排序	274
10.1	排序的基本概念	274

10.2	插入排序	275
10.2.1	直接插入排序	275
10.2.2	二分插入排序	277
10.2.3	希尔排序	277
10.3	交换排序	279
10.3.1	冒泡排序	279
10.3.2	快速排序	281
10.4	选择排序	285
10.4.1	直接选择排序	285
10.4.2	堆排序	287
10.5	归并排序	290
10.6	基数排序	293
10.7	各种内排序方法的比较和选择	296
	本章小结	297
	练习题 10	297
	上机实验题 10	298
第 11 章	外排序	299
11.1	外排序概述	299
11.2	磁盘排序	300
11.2.1	磁盘排序过程	300
11.2.2	多路平衡归并	301
11.2.3	初始归并段的生成	303
11.2.4	最佳归并树	305
11.3	磁带排序	307
11.3.1	多路平衡归并排序	307
11.3.2	多阶段归并排序	308
	本章小结	310
	练习题 11	310
	上机实验题 11	310
第 12 章	文件	311
12.1	文件的基本概念	311
12.1.1	什么是文件	311
12.1.2	文件的逻辑结构及操作	312
12.1.3	文件的存储结构	312
12.2	顺序文件	312
12.3	索引文件	313
12.3.1	ISAM 文件	314

12.3.2	VSAM 文件	317
12.4	哈希文件	319
12.5	多关键字文件	319
12.5.1	多重表文件	320
12.5.2	倒排文件	320
	本章小结	321
	练习题 12	321
	上机实验题 12	321
第 13 章	采用面向对象的方法描述算法	323
13.1	面向对象的概念	323
13.1.1	重要概念	324
13.1.2	主要优点	324
13.2	用 C++ 描述面向对象的程序	325
13.2.1	类	325
13.2.2	类对象	327
13.2.3	构造函数和析构函数	328
13.2.4	派生类	331
13.3	用 C++ 描述数据结构算法	334
13.3.1	顺序表类	334
13.3.2	链栈类	336
13.3.3	二叉树类	339
附录 A	综合实验题	343
附录 B	实验报告格式	345
附录 C	引用型参数的说明	346
附录 D	顺序表、顺序栈和顺序队列使用指针引用型参数的说明	348
附录 E	书中部分算法清单	351
附录 F	全国计算机专业数据结构考研大纲	354
	参考文献	356

绪 论

第 1 章

“数据结构”是计算机及相关专业的专业基础课之一，是一门十分重要的核心课程，主要学习用计算机实现数据组织和数据处理的方法。它也为计算机专业的后续课程(如操作系统、编译原理、数据库原理和软件工程等)的学习打下了坚实的基础。

另外，随着计算机应用领域的不断扩大，非数值计算问题占据了当今计算机应用的绝大多数，简单的数据类型已经远远不能满足需要，各数据元素之间的复杂联系已经不是普通数学方程式所能表达的了，无论设计系统软件还是应用软件都会用到各种复杂的数据结构。因此，掌握好数据结构课程的知识，对于提高解决实际问题的能力将会有很大的帮助。实际上，一个“好”的程序无非是选择一个合理的数据结构和好的算法，而好的算法的选择很大程度上取决于描述实际问题所采用的数据结构，所以，要想编写出“好”的程序，仅仅学习计算机语言是不够的，必须扎实地掌握数据结构的基本知识和基本技能。

1.1 什么是数据结构

在了解数据结构的重要性之后，开始讨论数据结构的定义，本节先从一个简单的学生表例子入手，继而给出数据结构的严格定义，接着分析了数据结构的几种类型，最后给出了数据结构和数据类型之间的区别与联系。

1.1.1 数据结构的定义

数据是人们利用文字符号、数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。例如，日常生活中使用的各种文字、数字和特定符号都是数据。从计算机的角度看，数据是所有能被输入到计算机中，且能被计算机处理的符号的集合。它是计算机操作的对象总称，也是计算机处理信息的某种特定的符号表示形式(例如，200902 班学生数据就是该班全体学生记录的集合)。

数据元素是数据(集合)中的一个“个体”,是数据的基本单位(例如,200902班中的每个学生记录都是一个数据元素),在计算机中通常被作为一个整体来进行考虑和处理。在某些情况下,数据元素也称为元素、结点、顶点、记录等。有时候,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数据最小单位,也称为字段或域(例如,200902班中每个数据元素即学生记录是由学号、姓名、性别和班号等数据项组成)。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。

数据结构是指数据以及相互之间的联系,可以看做是相互之间存在着某种特定关系的数据元素的集合,因此,可以把数据结构看成是带结构的数据元素的集合。数据结构包括以下几个方面。

(1) 数据元素之间的逻辑关系,即数据的逻辑结构。

(2) 数据元素及其关系在计算机存储器中的存储方式,即数据的存储结构,也称为数据的物理结构。

(3) 施加在该数据上的操作,即数据的运算。

数据的逻辑结构是从逻辑关系(主要是指相邻关系)上描述数据的,它与数据的存储无关,是独立于计算机的。因此,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。

数据的存储结构是逻辑结构用计算机语言的实现或在计算机中的表示(亦称为映像),也就是逻辑结构在计算机中的存储方式,它是依赖于计算机语言的。一般只在高级语言(例如,C/C++语言)的层次上来讨论存储结构。

数据的运算是定义在数据的逻辑结构之上的,每种逻辑结构都有一组相应的运算。例如,最常用的运算有检索、插入、删除、更新、排序等。数据的运算最终需在对应的存储结构上用算法实现。

因此,数据结构是一门讨论“描述现实世界实体的数学模型(通常为数值计算)及其之上的运算在计算机中如何表示和实现”的学科。

【例 1.1】 有一个学生表(数据)如表 1.1 所示。这个表中的数据元素是学生记录,每个数据元素由 4 个数据项(即学号、姓名、性别和班号)组成。讨论其存储结构。

表 1.1 学生表

学号	姓名	性别	班号
1	张斌	男	9901
8	刘丽	女	9902
34	李英	女	9901
20	陈华	男	9902
12	王奇	男	9901
26	董强	男	9902
5	王萍	女	9901

解 该表中的记录顺序反映了数据元素之间的逻辑关系,可以用学号标识每个学生记录,这种逻辑关系可以表示为: $\langle 1, 8 \rangle, \langle 8, 34 \rangle, \langle 34, 20 \rangle, \langle 20, 12 \rangle, \langle 12, 26 \rangle, \langle 26, 5 \rangle$ 。其中尖括号“ $\langle a_i, a_{i+1} \rangle$ ”表示元素 a_i 和 a_{i+1} 之间是相邻的,即 a_i 在 a_{i+1} 之前, a_{i+1} 在 a_i 之后。

这些数据在计算机存储器中的存储方式就构成了存储结构。通常可以采用 C/C++ 语言中的结构体数组和链表两种方式实现其存储结构。

存放上述学生表的结构体数组 Stud 定义如下：

```
struct
{ int no;           //存储学号
  char name[8];    //存储姓名
  char sex[2];     //存储性别
  char class[4];  //存储班号
} Stud[7] = {{1, "张斌", "男", "9901"}, ..., {5, "王萍", "女", "9901"}};
```

数组 Stud[] 中各元素在内存中顺序存放, 即 Stud[i] 存放在 Stud[i+1] 之前, 而 Stud[i+1] 存放在 Stud[i] 之后。

存放学生表的链表的结点类型 StudType 定义如下：

```
typedef struct studnode
{ int no;           //存储学号
  char name[8];    //存储姓名
  char sex[2];     //存储性别
  char class[4];  //存储班号
  struct studnode * next; //存储指向下一个学生的指针
} StudType;
```

学生表中每个学生记录采用一个 StudType 类型的结点存储, 一个学生结点的 next 域指向逻辑结构中它的后继学生对应的结点。从而构成一个链表, 其存储结构如图 1.1 所示, 首结点的指针为 head, 用它来标识该学生链表。由 head 所指结点的 next 域得到下一个结点的地址, 然后由它得到下一个结点的地址……如此这样可以找到任何一个结点的地址。

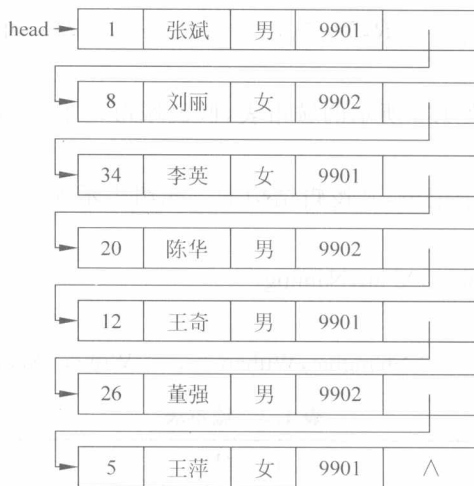


图 1.1 学生表的链表存储结构

对于学生表这种数据结构, 可以进行一系列的运算, 例如, 增加一个学生记录、删除一个学生记录、查找性别为“女”的学生记录、查找班号为 9902 的学生记录等。从前面介绍的两种存储结构看到, 同样的运算在不同的存储结构中其实现过程是不同的。