



普通高等教育“十一五”国家级规划教材

陈明 编著

# 数据结构

(C语言描述) (第2版)

21世纪  
计算机  
科学  
与  
技术  
实践  
型  
教  
程

丛书主编  
陈明

清华大学出版社





普通高等教育“十一五”国家级规划教材

陈明 编著

# 数据结构

## (C语言描述) (第2版)

常州大学图书馆  
藏书章

21世纪  
计算机  
科学与  
技术  
实践  
型教  
程

丛书主编  
陈明

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,主要包括算法、线性表、栈和队列、串、数组、树、图、查找、排序、递归和文件;为了加强对算法的理解,也介绍了算法分析方面的内容。本书语言精练、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。

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(C语言描述)/陈明编著. —2版. —北京:清华大学出版社,2011.11

(21世纪计算机科学与技术实践型教程)

ISBN 978-7-302-25680-9

I. ①数… II. ①陈… 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(2011)第103141号

责任编辑:谢琛 薛阳

责任校对:梁毅

责任印制:何芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦A座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954,jsjic@tup.tsinghua.edu.cn

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

印 刷 者:北京季蜂印刷有限公司

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:19.5

字 数:460千字

版 次:2011年11月第2版

印 次:2011年11月第1次印刷

印 数:1~4000

定 价:29.50元

---

产品编号:042053-01

# 《21 世纪计算机科学与技术实践型教程》

## 编辑委员会

主 任：陈 明

委 员：毛国君 白中英 叶新铭 刘淑芬 刘书家  
汤 庸 何炎祥 陈永义 罗四维 段友祥  
高维东 郭 禾 姚 琳 崔武子 曹元大  
谢树煜 焦金生 韩江洪

策划编辑：谢 琛

# 《21 世纪计算机科学与技术实践型教程》

## 序

21 世纪影响世界的三大关键技术：以计算机和网络为代表的信息技术；以基因工程为代表的生命科学和生物技术；以纳米技术为代表的新型材料技术。信息技术居三大关键技术之首。国民经济的发展采取信息化带动现代化的方针，要求在所有领域中迅速推广信息技术，导致需要大量的计算机科学与技术领域的优秀人才。

计算机科学与技术的广泛应用是计算机学科发展的原动力，计算机科学是一门应用科学。因此，计算机学科的优秀人才不仅应具有坚实的科学理论基础，而且更重要的是能将理论与实践相结合，并具有解决实际问题的能力。培养计算机科学与技术的优秀人才是社会的需要、国民经济发展的需要。

制订科学的教学计划对于培养计算机科学与技术人才十分重要，而教材的选择是实施教学计划的一个重要组成部分，《21 世纪计算机科学与技术实践型教程》主要考虑了下述两方面。

一方面，高等学校的计算机科学与技术专业的学生，在学习了基本的必修课和部分选修课程之后，立刻进行计算机应用系统的软件和硬件开发与应用尚存在一些困难，而《21 世纪计算机科学与技术实践型教程》就是为了填补这部分空白。将理论与实际联系起来，使学生不仅学会了计算机科学理论，而且也学会应用这些理论解决实际问题。

另一方面，计算机科学与技术专业的课程内容需要经过实践练习，才能深刻理解和掌握。因此，本套教材增强了实践性、应用性和可理解性，并在体例上做了改进——使用案例说明。

实践型教学占有重要的位置，不仅体现了理论和实践紧密结合的学科特征，而且对于提高学生的综合素质，培养学生的创新精神与实践能力有特殊的作用。因此，研究和撰写实践型教材是必需的，也是十分重要的任务。优秀的教材是保证高水平教学的重要因素，选择水平高、内容新、实践性强的教材可以促进课堂教学质量的快速提升。在教学中，应用实践型教材可以增强学生的认知能力、创新能力、实践能力以及团队协作和交流表达能力。

实践型教材应由教学经验丰富、实际应用经验丰富的教师撰写。此系列教材的作者不但从事多年的计算机教学，而且参加并完成了多项计算机类的科研项目，他们把积累的经验、知识、智慧、素质融合于教材中，奉献给计算机科学与技术的教学。

我们在组织本系列教材过程中，虽然经过了详细的思考和讨论，但毕竟是初步的尝试，不完善甚至缺陷不可避免，敬请读者指正。

本系列教材主编 陈明

2005 年 1 月于北京

# 前 言

数据是用来说明人类活动的事实观念或事物的一些文字、数字或符号。常用的数据类型分数值数据和非数值数据两大类。数值数据包括整数、定点数、浮点数等；非数值数据主要有逻辑数据、内码和交换码等。数据的级别由低向高依次为位、字节、字、数据项、数据字段、记录、文件、数据库等。

信息是指对某一特定的目的而言,具有意义的事实与知识,使源数据经系统的处理后成为决策或参考的依据。数据只是事实的记录,没有特定的目的;而信息则是针对某一问题来收集数据并进行处理,作为决策和参考的依据。通过数据处理,可将收集到的数据加以系统的处理,归纳出有价值的信息。常用的数据处理方式有编辑、排序、归并、分配、建档、更新、计算、链表、查找、查询等。

计算机科学是算法和算法变换的科学。数据结构主要是研究数据元素之间的关联方式,通常分为逻辑结构和物理结构两大类。同一逻辑结构可以对应不同的物理结构。程序存储是冯·诺依曼机的重要特征之一,构建计算机系统、利用计算机解决问题都是通过程序来实现。算法是求解问题的计算步骤的描述,算法是程序的核心和灵魂。算法的设计取决于数据的逻辑结构,而算法的实现依赖于指定的存储结构。在程序设计中,要从数据结构和算法两个方面考虑,才能得到高效而准确的结果。

在非数值计算中,处理对象已从简单数值发展到具有结构的数据,这就需要讨论如何有效地组织计算机的存储,并在此基础上有效地实现对象间的运算,数据结构就是研究与解决这些问题的重要基础。数据结构课程是人们在程序设计方面的经验总结,学会基本的程序设计,只能解决程序设计中的三成问题,而学会数据结构,却能解决程序设计中的八成问题。

数据结构课是计算机科学与技术专业的一门必修的、重要的核心基础课,是计算机程序设计的重要理论技术基础。通过数据结构课的学习,不仅可以使同学们掌握数据结构的基本内容、典型算法和使用方法,而且能够训练学生应用数据结构和算法进行具体应用问题的程序设计,进而提高程序设计能力。

本书介绍最常用的数据结构、各种数据结构的逻辑关系及其在计算机中的存储表示以及在数据结构上的运算等。主要内容包括算法、线性表、栈和队列、串、数组、树、图、查找、排序和文件等内容。

# 目 录

<b>第 1 章 绪论</b> .....	1
1.1 问题的提出 .....	1
1.2 基本术语 .....	2
1.3 数据结构的概念 .....	4
1.4 数据的逻辑结构 .....	6
1.5 数据的存储结构 .....	7
1.6 数据的运算 .....	9
1.7 数据的逻辑结构、存储结构及运算的关系.....	9
1.8 算法概述.....	10
1.8.1 算法与算法特性 .....	10
1.8.2 算法描述 .....	10
1.9 算法分析.....	11
小结 .....	13
习题 1 .....	14
<b>第 2 章 线性表</b> .....	15
2.1 线性表的定义与运算.....	15
2.1.1 线性表的定义 .....	15
2.1.2 线性表的运算 .....	16
2.2 线性表的顺序存储.....	19
2.2.1 顺序存储 .....	19
2.2.2 顺序结构线性表的运算 .....	20
2.2.3 顺序存储结构的优点 .....	23
2.2.4 顺序存储结构的缺点 .....	23
2.3 线性表的链式存储.....	23
2.3.1 线性链表 .....	23
2.3.2 线性链表的运算 .....	26
2.3.3 静态链表 .....	31

2.3.4	静态链表的运算 .....	31
2.3.5	循环链表 .....	32
2.3.6	循环链表的运算 .....	33
2.3.7	双向链表 .....	34
2.3.8	双向链表的运算 .....	35
2.3.9	链式存储结构的特点 .....	37
2.4	链式存储结构的应用 .....	37
2.4.1	约瑟夫问题 .....	37
2.4.2	一元多项式求和 .....	39
2.4.3	在集合方面的应用 .....	42
小结	.....	44
习题 2	.....	44
<b>第 3 章</b>	<b>栈和队列 .....</b>	<b>46</b>
3.1	栈 .....	46
3.1.1	栈的定义 .....	46
3.1.2	栈的顺序存储结构 .....	47
3.1.3	栈的链式存储结构 .....	51
3.1.4	顺序栈和链式栈的比较 .....	53
3.2	栈的应用 .....	53
3.2.1	迷宫问题 .....	53
3.2.2	算术表达式求值 .....	56
3.2.3	子程序的调用和返回 .....	59
3.2.4	数制转换 .....	60
3.3	队列 .....	61
3.3.1	队列的定义 .....	61
3.3.2	队列的顺序存储 .....	62
3.3.3	队列的链式存储 .....	68
3.3.4	优先队列 .....	72
3.4	队列的应用 .....	73
3.4.1	设备速度不匹配问题 .....	73
3.4.2	舞伴问题 .....	73
小结	.....	75
习题 3	.....	75
<b>第 4 章</b>	<b>串 .....</b>	<b>77</b>
4.1	串的基本概念 .....	77
4.2	串的存储结构 .....	78
4.2.1	串的静态存储结构 .....	79



4.2.2	串的动态存储结构 .....	80
4.3	串的基本运算及实现 .....	82
4.3.1	串的基本运算 .....	82
4.3.2	实现串的基本运算的算法 .....	83
4.4	模式匹配 .....	87
4.4.1	BF 算法 .....	87
4.4.2	KMP 算法 .....	90
	小结 .....	94
	习题 4 .....	94
<b>第 5 章</b>	<b>数组 .....</b>	<b>96</b>
5.1	数组的概念 .....	96
5.1.1	数组的定义及基本操作 .....	96
5.1.2	抽象数据类型数组的定义 .....	98
5.2	数组的顺序存储结构 .....	98
5.3	矩阵的压缩存储 .....	102
5.3.1	特殊矩阵的压缩存储 .....	102
5.3.2	稀疏矩阵的压缩存储 .....	104
	小结 .....	118
	习题 5 .....	118
<b>第 6 章</b>	<b>树 .....</b>	<b>120</b>
6.1	树 .....	120
6.1.1	树的定义 .....	120
6.1.2	树的表示方法 .....	121
6.1.3	树的基本术语 .....	121
6.1.4	树的 ADT 定义 .....	122
6.2	二叉树 .....	123
6.2.1	二叉树的定义及基本形态 .....	123
6.2.2	二叉树的存储结构 .....	125
6.2.3	二叉树的遍历 .....	127
6.3	线索二叉树 .....	130
6.3.1	二叉树的线索化 .....	131
6.3.2	利用线索遍历 .....	131
6.4	树、森林和二叉树的关系 .....	136
6.4.1	树的存储结构 .....	136
6.4.2	森林与二叉树的转换 .....	139
6.4.3	树和森林的遍历 .....	142

6.5 哈夫曼树及其应用 .....	143
6.5.1 与哈夫曼树有关的定义 .....	143
6.5.2 哈夫曼树的构造 .....	145
6.5.3 哈夫曼树的应用 .....	146
小结 .....	151
习题 6 .....	151
<b>第 7 章 图</b> .....	<b>155</b>
7.1 图的基本概念 .....	155
7.2 图的存储结构 .....	159
7.2.1 邻接矩阵 .....	160
7.2.2 邻接表 .....	162
7.2.3 十字链表 .....	166
7.2.4 邻接多重表 .....	167
7.3 图的遍历 .....	169
7.3.1 深度优先搜索 .....	169
7.3.2 广度优先搜索 .....	172
7.4 生成树 .....	174
7.4.1 普里姆(Prim)算法 .....	175
7.4.2 克鲁斯卡尔(kruskal)算法 .....	178
7.5 最短路径 .....	180
7.5.1 单源最短路径 .....	180
7.5.2 顶点之间的最短路径 .....	183
7.6 拓扑排序 .....	184
7.7 关键路径 .....	188
小结 .....	195
习题 7 .....	195
<b>第 8 章 查找</b> .....	<b>199</b>
8.1 基本概念 .....	199
8.2 线性表的查找 .....	200
8.2.1 顺序查找 .....	200
8.2.2 折半查找 .....	202
8.2.3 分块查找 .....	205
8.3 树表的查找 .....	208
8.3.1 二叉查找树 .....	208
8.3.2 平衡二叉树 .....	214
8.3.3 B-树 .....	220

8.4 哈希表的查找 .....	222
8.4.1 哈希表 .....	222
8.4.2 构造哈希表的基本方法 .....	223
8.4.3 解决冲突的方法 .....	225
8.4.4 哈希表的查找方法 .....	228
8.5 查找方法的比较 .....	228
小结 .....	229
习题 8 .....	229
<b>第 9 章 排序</b> .....	<b>232</b>
9.1 基本概念 .....	232
9.2 内部排序方法 .....	235
9.2.1 插入排序 .....	235
9.2.2 冒泡排序 .....	239
9.2.3 快速排序 .....	240
9.2.4 选择排序 .....	243
9.2.5 归并排序 .....	248
9.2.6 基数排序 .....	251
9.3 内部排序方法比较 .....	256
9.4 外部排序方法简介 .....	257
小结 .....	257
习题 9 .....	258
<b>第 10 章 递归</b> .....	<b>261</b>
10.1 递归的定义 .....	261
10.2 典型递归问题 .....	262
10.2.1 汉诺塔问题 .....	262
10.2.2 八皇后问题 .....	264
10.3 递归的实现 .....	266
10.4 递归转化为非递归的一般过程 .....	270
10.5 递归的时间和空间复杂度 .....	273
小结 .....	274
习题 10 .....	275
<b>第 11 章 文件</b> .....	<b>276</b>
11.1 外存储器的介绍 .....	276
11.2 有关文件的概念 .....	277
11.2.1 文件及其类别 .....	278

11.2.2 文件的操作.....	279
11.3 文件的组织.....	280
11.3.1 顺序文件.....	281
11.3.2 索引文件.....	282
11.3.3 散列文件.....	287
11.3.4 多关键字文件.....	289
小结.....	291
习题 11 .....	291
附录 上机实验.....	293
参考文献.....	296

# 第 1 章 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、什么是数据结构及数据结构的一些相关基本概念等。这对于深刻理解书中后面章节的内容将会有很大的帮助。

## 1.1 问题的提出

在计算机发展的初期,计算机主要用于数值计算。由于计算机的存储容量及计算速度的限制,程序设计人员把主要精力集中在程序设计技巧的研究上,但随着计算机应用领域的扩大和软硬件技术的发展,计算机对信息的处理已从单一的数值计算扩展到解决非数值运算问题,能够处理的信息也由简单的数值扩展到字符、图像、声音等具有复杂结构的数据。“数据结构”就是随着计算机的产生和发展而出现的一门计算机课程。

非数值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。目前,系统程序和许多应用程序对数据的运算处理规模很大,并且数据的结构也非常复杂,因此,处理这些问题的关键点已不再仅是考虑数学分析和计算方法,而是需要设计出合理的数据结构,这样才能有效地解决问题。

计算机科学是一门研究如何用计算机进行信息表示和处理的科学,涉及信息表示和信息处理两个问题。而信息的表示和组成又直接关系到处理信息的程序的效率及效果。随着计算机相关技术应用的扩充,信息量逐渐增加,信息范围也逐渐拓宽,使许多系统程序和应用程序的规模变大,结构变得相当复杂。这就要求计算机程序对待处理的数据对象进行系统的研究,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。

### 例 1-1 电话号码查询问题。

编写一个查询某个城市或单位的私人电话号码的程序。要求对任意给出的一个姓名,若该人有电话号码,则迅速找到其电话号码,否则指出该人没有电话号码。

要解决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。

要想迅速地查找某人的电话号码,必须根据表中信息的结构和特点,确定存储方式并写出好的查找算法。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,顺序查找就不实用了。若这张表是按姓氏排列的,则可另制一张姓氏索引表,采用如图 1-1 所示

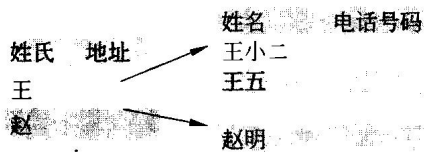


图 1-1 具有索引表的存储结构

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须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。

“数据结构”作为计算机专业的核心课程之一,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,要想更有效地使用计算机来处理实际问题,就必须打好数据结构基础。

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,这个等式描述了算法、数据结构和程序之间的关系,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合理的数据结构,加之设计一些应用在其上的高效的算法,而且算法的好坏很大程度上取决于描述实际问题的数据结构。

## 1.2 基本术语

为了更好地理解数据结构的概念,先将数据结构中的一些常用名词和术语给出解释。

### 1. 数据

数据(Data)作为信息的载体,是对客观事物的描述,这种描述形式可以是数字、字符以及所有能输入到计算机中被计算机程序识别、处理的信息的集合。数据不仅是数学领域中的整数和实数等。在计算机领域数据是对客观事物的进一步抽象,数据所能表述的范畴是计算机可以处理的字符串、图像、声音等各种信息。如表 1-1 所示,张风的英语成绩为 92 分,92 就是该同学的英语成绩数据。

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

数据项(Data Item)是具有独立意义的不可分的最小数据单位,它是对数据结构中的数据元素属性的抽象描述。数据项也被称为字段或域,参见图 1-2。

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

图 1-2 数据元素和数据项

## 3. 数据元素

数据元素(Data Element)是数据的基本单位,是对一个客观实体的数据形式的抽象描述。一个数据元素可以由若干个数据项组成。数据元素也被称为结点或记录。

利用表 1-1 的例子来说明数据项和数据元素,整个表描述的是学生的成绩数据,一个学生的某一门成绩就是其中的一个数据项。

一个学生的成绩是一个数据元素,那么这个数据元素由学号、姓名、语文成绩、数学成绩、英语成绩 5 个数据项组成。

## 4. 数据对象

具有相同性质的数据元素的集合就是一个数据对象(Data Object),它是数据的一个子集。如图 1-2 所示,一个班级的成绩表可以看作一个数据对象。又例如,集合 $\{1, 2, 3, 4, 5, \dots\}$ 是自然数的数据对象,而集合 $\{'a', 'b', 'c', 'd', \dots, 'z'\}$ 是英文字母表的数据对象。可以看出,数据对象可以是无限的,也可以是有限的。

## 5. 数据类型

数据类型(Data Type)是将客观事物集抽象描述成具有相同性质的数据元素的集合以及对这个数据集合中数据元素的一组操作的整体表述。例如在 C 语言中整数类型是集合 $C = \{0, \pm 1, \pm 2, \pm 3, \pm 4, \dots\}$ 及定义在该集合上的加、减、乘、整除和取余等一组操作。

每个数据项属于某个确定的数据类型,数据类型分为原子类型和结构类型。

### 1) 原子类型

如果一个数据元素由一个数据项组成,这个数据元素的类型就是这个数据项的数据类型,值在逻辑上不可分解。

### 2) 结构类型

如果由多个不同类型的数据项组成,则这个数据元素的类型就是由各数据项类型构造而成的结构类型,值由若干成分按某种结构组成。上面提到的学生成绩表中,数据项姓名的数据类型为字符型,而成绩的数据类型是数值型,所以这个数据元素是一个结构类型。上述成绩表数据可用 C 语言的结构体数组 `Class1stu[50]` 来存储。

```
struct Stu
{ /* 数据项 */
    int stuID;
    char name[20];
    int maths_score;
    int chinese_score;
    int english_score;
} Class1stu[50];
```

不同的高级语言提供的基本数据类型有所不同,在 C 语言中,提供了实型、整型、字符型和指针型等基本数据类型。

## 6. 抽象数据类型

抽象数据类型 (Abstract Data Type) 就是对象的数学模型,简称为 ADT,是用户在数据类型基础上新定义的数据类型。抽象数据类型包含数据组成结构的定义以及对建立在其上的处理操作的描述,可见抽象数据类型是数据和数据使用者的一个接口。调用者通过 ADT 中定义的接口来创建、销毁及控制 ADT 中的对象以期满足调用者的需要。

抽象数据类型主要描述建立在抽象层次相对较高的数据相关的逻辑特性,而不关心其在不同的计算机程序语言环境中的表示和实现,只要维持数据的抽象数学特性,数据的使用方式将保持不变。从抽象数据类型的角度入手设计软件系统可大大提高软件构件的复用率。抽象数据类型的定义包括数据对象定义、数据关系定义和基本操作定义三部分,其书写格式为:

ADT——抽象数据类型名

- (1) 数据对象:对元数据对象结构的定义;
- (2) 数据关系:对数据对象之间关系的定义;
- (3) 基本操作:建立在数据对象及数据关系上的相关基本操作的定义。

抽象数据类型可以看作是描述问题的模型,它独立于具体实现。它的优点是将数据和操作封装在一起,使得用户程序只能通过 ADT 里定义的某些操作来访问其中的数据,从而实现了信息隐藏。

## 1.3 数据结构的概念

数据结构是指数据之间的相互关系(即数据的组织形式)及建立在这些数据上的运算方法的集合。为了进一步理解数据结构,举一个简单的例子来说明。

如表 1-2 所示的学生基本情况表,记录了某校全体学生的姓名和相应的基本信息,现在要求设计一个算法,当给定任何一个学生的姓名时,计算机能够查出该学生的基本信息,如果根本就不存在这个学生,计算机就输出“无此学生记录!”。



表 1-2 学生基本情况表

编 号	姓 名	年 级	年 龄	性 别
01	张风	1 年级	12	男
02	李强	1 年级	14	男
03	林海	2 年级	14	男
04	李南	2 年级	10	男
05	韩凤	3 年级	11	女
06	赵加	1 年级	12	女
⋮	⋮	⋮	⋮	⋮

这个例子实现的是查找功能。查找算法的设计完全依赖于基本情况表中学生姓名和相应信息在计算机内的存储方式。

如果学生基本情况表中的学生的姓名是随意排列的,排列次序没有任何规律,如表 1-2 所示,那么在给定一个学生姓名时,则只能对学生的基本情况表从头到尾逐个与给定的姓名比较,顺序查找直至找到所给的姓名为止,很有可能表中根本就不存在这个人。虽然这种方法很简单,是线性查询,但是这会浪费很多时间,效率低。

如果将基本情况表按字母顺序排列学生的姓名和相应的情况,如表 1-3 所示,并且再构造一个字母索引表(表 1-4),这个表用来登记以某个字母开头的第一个学生姓名在基本情况表中的起始位置。当查找某学生的情况时,先从索引表中查到以该字母开头的第一个学生姓名在基本情况表中的起始位置,然后,从此起始处开始查找,而不必去查看以其他字母开头的学生记录,通过建立这样一种数据组织形式,查找的效率就会有很大的提高。另外,还可以按年级进行排序,然后建立一个年级的索引表,当查询某个年级的学生

表 1-3 按开头字母排序的基本情况表

编 号	姓 名	年 级	年 龄	性 别
⋮	⋮	⋮	⋮	⋮
11	韩凤	3 年级	11	女
⋮	⋮	⋮	⋮	⋮
24	李强	1 年级	14	男
25	李南	2 年级	10	男
26	林海	2 年级	14	男
⋮	⋮	⋮	⋮	⋮
87	张风	1 年级	12	男
88	赵加	1 年级	12	女
⋮	⋮	⋮	⋮	⋮

表 1-4 字母索引表

开 头 字 母	编 号	开 头 字 母	编 号
⋮	⋮	⋮	⋮
H	11	Z	87
⋮	⋮	⋮	⋮
L	24		