

北京市教育委员会共建项目专项资助  
高等学校计算机教育“十二五”规划教材

# 数据结构 (C语言版)

陈 明 编著

HUJU JIEGOU

(C YUYANBAN)

中国铁道出版社  
CHINA RAILWAY PUBLISHING HOUSE

---

清华大学出版社  
北京

# 数据结构 (英文版)

第 2 版

清华大学出版社  
北京

第 2 版

北京市教育委员会共建项目专项资助  
高等学校计算机教育“十二五”规划教材

# 数 据 结 构

## (C 语言版)

陈 明 编著

## 内 容 简 介

本书系统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、树、图、查找、排序和递归等。为了加强对算法的理解, 还介绍了算法分析方面的内容。

本书选材精练、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, 并都附有丰富的习题, 便于学生理解与掌握。

本书既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用书。

### 图书在版编目(CIP)数据

数据结构: C语言版 / 陈明编著. —北京: 中国铁道出版社, 2012. 11

北京市教育委员会共建项目专项资助 高等学校计算机教育“十二五”规划教材  
ISBN 978-7-113-15621-3

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. ①TP311.12②TP312

中国版本图书馆CIP数据核字(2012)第259050号

书 名: 数据结构(C语言版)  
作 者: 陈明 编著

---

策 划: 秦绪好 读者热线: 400-668-0820  
责任编辑: 赵 鑫  
编辑助理: 赵 迎  
封面设计: 刘 颖  
封面制作: 白 雪  
责任印制: 李 佳

---

出版发行: 中国铁道出版社(100054, 北京市西城区右安门西街8号)  
网 址: <http://www.51eds.com>  
印 刷: 航远印刷有限公司  
版 次: 2012年11月第1版 2012年11月第1次印刷  
开 本: 787 mm×1 092 mm 1/16 印张: 12.75 字数: 300千  
印 数: 1~3 000册  
书 号: ISBN 978-7-113-15621-3  
定 价: 25.00元

---

版权所有 侵权必究

凡购买铁道版图书, 如有印制质量问题, 请与本社教材图书营销部联系调换。电话: (010) 63550836

打击盗版举报电话: (010) 63549504

高等学校计算机教育“十二五”规划教材

主 任：陈 明

副主任：叶曲炜 严晓舟

委 员：（按姓氏笔画排序）

王希更	王智广	王锁柱	刘贵龙	李 环
李海生	李素朵	张晓明	陈志泊	陈晓云
赵裕民	郝 莹	侯耀军	姚 琳	秦绪好
袁东光	袁 薇	徐孝凯	徐 慧	郭成芳
曹永存	解 凯	管建和		

随着计算机科学与技术的飞速发展，现代计算机系统的功能越来越强大，应用也越来越广泛。尤其是快速发展的计算机网络，它不仅是连接计算机的桥梁，而且已成为扩展计算能力、提供公共计算服务的平台。计算机科学对人类社会的发展做出了卓越的贡献。

计算机科学与技术的广泛应用是推动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，计算机学科的优秀创新人才不仅应具有坚实的理论基础，还应具有将理论与实践相结合来解决实际问题的能力。培养计算机学科的创新人才是社会的需要，是国民经济发展的需要。

计算机学科的发展呈现出学科内涵宽泛化、分支相对独立化、社会需求多样化、专业规模巨大化和计算教育大众化等特点。一方面，使得计算机企业成为朝阳企业，软件公司、网络公司等 IT 企业的数量和规模越来越大；另一方面，对计算机人才的需求规格也发生了巨大变化。在大学中，单一计算机精英型教育培养的人才已不能满足实际需要，社会需要大量的具有职业特征的计算机应用型人才。

计算机应用型教育的培养目标可以用“知识、能力和素质”这三个基本要素来描述。

知识是基础、载体和表现形式，从根本上影响着能力和素质。学习知识的目的是获得能力和不断地提升能力。能力和素质的培养必须通过知识传授来实现，能力和素质也必须通过知识来表现。

能力是核心，是人才特征的最突出的表现。计算机学科人才应具备计算思维能力、算法设计与分析能力、程序设计能力和系统能力（系统的认知、设计、开发和应用）。计算机应用型教育对人才培养的能力要求主要包括应用能力和通用能力。应用能力主要是指用所学知识解决专业实际问题的能力；通用能力表现为跨职业能力，并不是具体的专业能力和职业技能，而是对不同职业的适应能力。计算机应用型教育培养的人才所应具备的三种通用能力是学习能力、工作能力、创新能力。

素质是指具有良好的公民道德和职业道德，具有合格的政治思想素养，遵守计算机法规和法律，具有人文、科学素养和良好的职业素质等。计算机应用型人才素质主要是指工作的基本素质，且要求在从业中必须具备责任意识，能够将自己职责范围内的工作认真负责地完成。

计算机应用型教育课程类型分为通用课程、专业基础课程、专业核心课程、专业选修课程、应用课程、实验课程、实践课程。课程是载体，是实现培养目标的重要手段。教育理念的实现必须借助于课程来完成。本系列规划教材的特点是重点突出，理论够用，注重应用，内容先进实用。

本系列教材恐有不足之处，敬请各位专家、老师和广大同学指正。



2012年3月

“数据结构”是计算机类专业的一门必修的、重要的核心基础课。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。

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一定结构的数据，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，学生不仅可以掌握数据结构的基本内容、典型算法和使用方法，而且能够训练应用数据结构和算法进行具体应用问题的程序设计。

本书分为 10 章，介绍最常用的数据结构、各种数据结构的逻辑关系、在计算机中的存储表示，以及在数据结构上的运算等。主要内容包括线性表、栈和队列、串、数组、树、图、查找、排序和递归等内容。

本书在结构上呈积木式，注重实践应用，各种常用数据结构的介绍从实际出发，避免抽象的理论论述和复杂的公式推导，典型的算法介绍深入浅出、简洁明了。每章都设有小结和习题。通过这些习题的练习，不仅能加深学生对基本概念和定义的理解，还能通过上机提高编程能力和程序调试能力。

由于编者水平有限，书中不足之处在所难免，敬请读者批评指正。

陈 明  
2012 年 10 月



<b>第 1 章 概论</b> .....	<b>1</b>
1.1 基本概念与术语 .....	1
1.2 数据结构的概念 .....	3
1.3 数据的逻辑结构 .....	5
1.3.1 数据结构的形式化定义 .....	5
1.3.2 数据的逻辑结构类型 .....	5
1.4 数据的存储结构 .....	6
1.4.1 顺序存储方法 .....	6
1.4.2 链式存储方法 .....	6
1.4.3 索引存储方法 .....	7
1.4.4 散列存储方法 .....	7
1.5 数据的运算 .....	7
1.6 算法与算法特性 .....	8
1.6.1 算法及其特性 .....	8
1.6.2 算法的描述方法 .....	8
1.6.3 算法与程序及数据结构 .....	10
1.7 算法性能分析 .....	10
1.8 算法度量 .....	11
1.8.1 算法时间复杂度 .....	11
1.8.2 复杂度函数的增长率 .....	13
小结 .....	15
习题 1 .....	15
<b>第 2 章 线性表</b> .....	<b>16</b>
2.1 线性表定义及其运算 .....	16
2.1.1 线性表 .....	16
2.1.2 线性表的运算 .....	17
2.2 线性表的顺序存储 .....	19
2.2.1 顺序存储结构 .....	19
2.2.2 顺序结构线性表的运算 .....	20
2.2.3 顺序存储结构的特点 .....	22
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 循环链表 .....	29
2.3.4 循环链表的运算 .....	29
2.3.5 双向链表 .....	30
2.3.6 双向链表的运算 .....	31

2.3.7 链式存储结构的特点.....	32
2.4 链式存储结构的应用.....	33
小结.....	36
习题 2.....	36
<b>第 3 章 栈和队列.....</b>	<b>38</b>
3.1 栈.....	38
3.1.1 栈的定义及其运算.....	38
3.1.2 栈的顺序存储结构及其运算的实现.....	39
3.1.3 栈的链式存储结构.....	43
3.2 栈的应用.....	44
3.2.1 子程序的调用问题.....	44
3.2.2 算术表达式求值.....	44
3.3 队列.....	47
3.3.1 队列的定义.....	47
3.3.2 队列的顺序存储.....	47
3.3.3 队列的链式存储.....	52
3.4 队列的应用.....	55
小结.....	56
习题 3.....	56
<b>第 4 章 串.....</b>	<b>58</b>
4.1 串的基本概念.....	58
4.2 串的存储结构.....	59
4.2.1 串的静态存储结构.....	59
4.2.2 串的动态存储结构.....	61
4.3 串的基本运算.....	62
4.3.1 常见的基本运算.....	62
4.3.2 实现串的基本运算的算法.....	63
4.4 模式匹配.....	65
4.5 串在文本编辑中的应用.....	67
小结.....	68
习题 4.....	68
<b>第 5 章 数组.....</b>	<b>70</b>
5.1 数组及其基本操作.....	70
5.1.1 数组的概念.....	70
5.1.2 数组的基本操作.....	72
5.2 数组的存储结构.....	72
5.3 数组在矩阵运算中的应用.....	75
5.3.1 特殊矩阵的压缩存储.....	75
5.3.2 稀疏矩阵的压缩存储.....	77
小结.....	83
习题 5.....	83

第 6 章 树 .....	85
6.1 树 .....	85
6.1.1 树的定义 .....	85
6.1.2 树的常用术语 .....	86
6.1.3 树的基本操作 .....	87
6.2 二叉树 .....	87
6.2.1 二叉树的定义 .....	87
6.2.2 二叉树的存储结构 .....	89
6.2.3 二叉树的遍历 .....	91
6.2.4 二叉树遍历的应用 .....	93
6.3 线索二叉树 .....	94
6.4 树、森林和二叉树的关系 .....	97
6.4.1 树的存储结构 .....	97
6.4.2 森林与二叉树的转换 .....	100
6.4.3 树和森林的遍历 .....	102
6.5 哈夫曼树 .....	103
6.5.1 哈夫曼树的定义 .....	103
6.5.2 哈夫曼树的构造 .....	104
小结 .....	105
习题 6 .....	105
第 7 章 图 .....	108
7.1 图的概念及其操作 .....	109
7.1.1 图的概念 .....	109
7.1.2 图的基本操作 .....	113
7.2 图的存储结构 .....	113
7.2.1 邻接矩阵 .....	113
7.2.2 邻接表 .....	115
7.2.3 十字链表 .....	118
7.2.4 邻接多重表 .....	119
7.3 图的遍历 .....	121
7.3.1 深度优先搜索 .....	121
7.3.2 广度优先搜索 .....	123
7.4 图的应用 .....	124
7.4.1 生成树 .....	124
7.4.2 最短路径 .....	128
7.4.3 拓扑排序 .....	131
7.5 关键路径法 .....	133
小结 .....	139
习题 7 .....	139

第 8 章 查找 .....	142
8.1 基本概念 .....	142
8.2 线性表的查找 .....	143
8.2.1 顺序查找 .....	143
8.2.2 折半查找 .....	145
8.2.3 分块查找 .....	147
8.3 二叉查找树 .....	148
8.4 哈希表的查找 .....	152
8.4.1 哈希表 .....	152
8.4.2 构造哈希表的基本方法 .....	153
8.4.3 解决冲突的方法 .....	154
8.5 各种查找方法的比较 .....	156
小结 .....	157
习题 8 .....	157
第 9 章 排序 .....	159
9.1 基本概念 .....	159
9.2 内部排序 .....	161
9.2.1 插入排序 .....	161
9.2.2 冒泡排序 .....	164
9.2.3 快速排序 .....	166
9.2.4 选择排序 .....	168
9.2.5 归并排序 .....	173
9.3 内部排序方法比较 .....	176
小结 .....	177
习题 9 .....	177
第 10 章 递归 .....	179
10.1 递归的定义与类型 .....	179
10.1.1 递归的定义 .....	179
10.1.2 递归的类型 .....	179
10.2 递归应用举例 .....	180
10.2.1 汉诺塔问题 .....	180
10.2.2 八皇后问题 .....	182
10.3 递归的实现 .....	183
10.4 递归到非递归的转换过程 .....	187
10.5 递归的时间和空间复杂度 .....	190
小结 .....	191
习题 10 .....	191
参考文献 .....	192

# 第1章

## 概论

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、数据结构的定义及数据结构的相关概念等。这对深刻理解后面章节的内容会有很大帮助。

计算机发展初期，人们使用计算机的目的主要是处理数值的计算问题，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，计算机对信息的处理加工已从单一的数值计算发展到大量地解决非数值问题，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。随着计算机数据的复杂化，数据结构这门学科产生并发展起来。

非数值问题的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学和计算方法，而是研究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。这里涉及两个问题：信息的表示和信息的处理。而信息的表示和组成又直接关系到处理信息程序的效率。随着计算机的普及、信息量的增加、信息范围的拓宽，许多系统程序和应用程序的规模越来越大，结构也相当复杂，这就要求人们对计算机程序加工的对象进行系统研究，即研究数据特性及数据之间的关系，而数据结构正是描述数据特性及数据之间关系的一门课程。

数据结构是计算机专业的核心课程之一，在众多计算机系统软件和应用软件中都要用到各种数据结构。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难以完成众多复杂的研究课题，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，这个等式形象地描述了算法、数据结构和程序之间的关系，这里的数据结构是指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适的数据结构，加之设计一个好的算法，而且好的算法很大程度上取决于实际问题的数据结构。

## 1.1 基本概念与术语

为了更好地理解数据结构，首先介绍数据结构中的常用概念与术语。

## 1. 数据

数据 (Data) 是信息的载体, 它是描述客观事物的数、字符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。数据不仅是通常意义下的整数和实数, 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、声音等都可以称为数据, 所以不能只是泛泛地理解数据这个概念。下面进一步解释数据的定义, 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. 数据元素

数据元素 (Data Element) 是数据的单位, 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数据项组成。数据元素也被称为结点或记录。

## 3. 数据项

数据项 (Data Item) 是数据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、域。

利用表 1-1 所示的例子来说明数据项和数据元素, 整个表记录的是学生的成绩数据, 每个学生的一条记录 (包括学号、姓名、语文成绩、数学成绩、英语成绩) 就是其中的一个数据元素, 而学号、姓名、语文、数学、英语就是数据项, 如图 1-1 所示。



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 1-1 数据元素和数据项

## 4. 数据对象

数据对象 (Data Object) 是指具有相同性质的数据元素的集合, 它是数据的一个子集。如表 1-1 所示的例子中, 一个班级的成绩表可以看做一个数据对象。例如, 集合 {1,2,3,4,5,...} 是自然数的数据对象, 而集合 {'a', 'b', 'c', 'd', ..., 'z'} 是英文字母表的数据对象。可以看出, 数据对象可以是有限的, 也可以是无限的。

## 5. 数据类型

数据类型 (Data Type) 是具有相同性质的计算机数据的集合及定义在这个数据集合上的一组操作的总称。例如, C 语言中的整数类型是集合  $C=\{0, \pm 1, \pm 2, \pm 3, \pm 4, \dots\}$  及定义在该集合上的加、减、乘、整除和取余等一组操作。数据类型封装 (抽象) 了数据存储与操作的具体细节。

每个数据项属于某个确定的基本数据类型, 数据类型的种类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，在逻辑上其值不可分解。例如，`int i` 表示整型，`float j` 表示浮点型等。

### (2) 结构类型

如果由多个不同的类型的数据项组成，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。例如 `struct stu`，即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];
```

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

## 1.2 数据结构的概念

前面解释了数据结构中的一些重要术语，现在对数据结构这个概念进行归纳，数据结构就是数据之间的相互关系（即数据的组织形式）及在这些数据上定义的数据运算方法的集合，一般包括以下 3 方面的内容：

- ① 数据之间的逻辑关系，又称数据的逻辑结构。
- ② 数据元素及其关系在计算机存储器内的表示，称为数据的存储结构，即物理结构。
- ③ 数据的运算，即对数据进行的操作。

为了进一步理解数据结构，举一个简单的例子来说明。

有一个学生的基本情况表，排列顺序没有任何规律，如表 1-2 所示。表中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，如果不存在这个学生，计算机就输出“无此学生记录!”。

这个例子实现的是查找功能。可以看出，这个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。

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	6	男
02	李强	1 年级	6	男
03	林海	2 年级	7	男
04	李南	2 年级	7	男
05	韩凤	3 年级	8	女
06	赵加	1 年级	6	女
⋮	⋮	⋮	⋮	⋮

如果学生基本情况表中学生的姓名是随意排列的,那么在给定一个学生姓名时,只能对学生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,很有可能查找完全部基本情况表还没有找到这个人。虽然这种方法很简单,但采用线性穷举式查询会浪费很多时间,效率低。

如果将基本情况表进行适当地组织,按字母顺序排列学生的姓名和相应的情况,如表 1-3 所示。再构造一个字母索引表,如表 1-4 所示,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的记录。通过建立这样一种数据组织形式,查找效率会大大提高。此外,还可以按年级进行排序,然后建立一个年级索引表,当查询某个年级的学生时,可以先找到这个年级所在的开始位置,然后再查询,这样就提高了查找速度。

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	8	女
∴	∴	∴	∴	∴
24	李强	1 年级	6	男
25	李南	2 年级	7	男
26	林海	2 年级	7	男
∴	∴	∴	∴	∴
87	张风	1 年级	6	男
88	赵加	1 年级	6	女
∴	∴	∴	∴	∴

表 1-4 字母索引表

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

对于不同的存储结构,要构造出完全不同的算法。算法和数据结构是密切相关的,算法依赖于具体的数据结构,数据结构也直接关系到算法的选择和效率。

此外,当新生入校时,就需要将新生的姓名和相关信息添加到学生基本情况表中;当学生毕业或转学时,应从基本信息表中删除该学生的记录。这就要求在已安排好的结构上进行插入和删除操作。除此之外,还可能对学生基本情况表进行修改等运算。这些运算由计算机完成,就需要设计相应的算法,也就是说,数据结构还需要给出每种结构中的各种运算和算法。

通过前面的介绍,可以直观认为:数据结构是研究数据元素之间的相互关系和这种关系在计算机中的存储表示,并对这种结构定义相应的运算,设计出相应的算法,而且确保经过这些运算后所到的结果仍然是原来的结构类型。

数据结构讨论的问题主要有:如何以最节省存储空间的方式来表示数据和各种不同的数据结构表示方法及其相关算法;如何有效地改进算法效率使程序执行速度更快;数据处理的各种技巧,如排序、查找等算法的介绍等。

数据结构包括逻辑结构和存储结构(物理结构)。逻辑结构是在逻辑关系上描述数据的,它与数据在计算机内的存储方式没有关系,可看做从具体问题中抽象出来的数据模型。存储结构是指数据结构在计算机内的表示,又称物理结构。存储结构分有顺序存储结构和链式存储结构。



## 1.3 数据的逻辑结构

### 1.3.1 数据结构的正式化定义

数据结构在形式上可定义为一个二元素组：

$$\text{Data\_Structure}=(D,R)$$

其中， $D$  是数据元素的有限集合， $R$  是  $D$  上关系的有限集合。

由此可以看出，数据结构由两部分构成：

- ① 数据元素的集合  $D$ 。
- ② 数据元素之间关系的集合。

假设要设计一个事务管理的程序，用来管理计算机系研究课题小组的各项事务。现在需要设计一个数据结构，如果要求每个课题组由 1 名教授、1~4 名研究生和 1~8 名本科生组成，在小组中，1 名教授指导 1~4 名研究生，每名研究生指导 1~2 名本科生，得到一个数据结构：

$$\text{Group}=(P,R)$$

其中， $P$  表示数据元素，包括教授、研究生、本科生，即  $P=\{T,G_1,\dots,G_n,S_{11},S_{12},\dots,S_{nm}\}$  ( $1\leq n\leq 4$ ,  $1\leq m\leq 2$ )。  $T$  表示教授、 $G_i$  表示研究生、表示  $S_{ij}$  本科生。

$R$  表示小组成员的关系，其关系有两种：

- ① 教授和研究生： $R1=\{<T,G_i> \mid 1\leq i\leq n, 1\leq n\leq 4\}$ ;
- ② 研究生和本科生： $R2=\{<G_i,S_{ij}> \mid 1\leq i\leq n, 1\leq j\leq m, 1\leq n\leq 4, 1\leq m\leq 2\}$ 。

再举一个例子，一周 7 天的数据结构可表示为

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

$D=\{\text{星期一,星期二,星期三,星期四,星期五,星期六,星期天}\}$

$S=\{<\text{星期天,星期一}>, <\text{星期一,星期二}>, <\text{星期二,星期三}>, <\text{星期三,星期四}>, <\text{星期四,星期五}>, <\text{星期五,星期六}>, <\text{星期六,星期天}>\}$

以上数据结构用图表示为图 1-2 所示的形式。

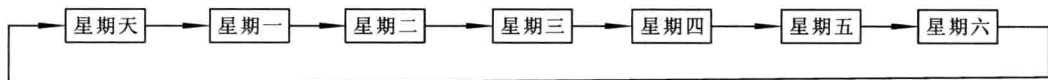


图 1-2 1 周 7 天数据结构图示

总而言之，数据结构是相互之间存在一种或多种特定关系的数据元素的集合，这个关系描述的是数据元素之间的逻辑关系。数据的逻辑关系又称数据的逻辑结构，它与数据的存储无关，因此，数据的逻辑结构可以看做从具体的问题中抽象出来的数学模型。

### 1.3.2 数据的逻辑结构类型

数据的逻辑结构分为 3 种典型结构：集合、线性结构和非线性结构。

① 集合的特征是元素间为松散的关系，只是元素属于某个集合而已。例如，各种颜色属于色彩集合。

② 线性结构的逻辑特征是有且仅有一个起始结点和一个终端结点，并且所有结点只有一个直接前驱和一个直接后继，如线性表、队列等。结点之间是一一对一的关系，如前面的学生成绩表中每个数据元素之间的关系。

③ 非线性结构的特征是一个结点可能有多个直接前驱或多个直接后继，如树、图等。树的结点之间是一对多的关系，图的结点之间的关系是多对多的关系。