

博雅

高等学校计算机技术类课程规划教材

# 数据结构与算法

林 劫 刘 震 波  
陈端兵 戴 波 ◎主 编

[ Data Structures  
and Algorithms ]



北京大学出版社  
PEKING UNIVERSITY PRESS



高等学校计算机技术类课程规划教材

# 数据结构与算法

主 编 林 劼 刘 震 陈端兵 戴 波



北京大学出版社  
PEKING UNIVERSITY PRESS

## 图书在版编目 (CIP) 数据

数据结构与算法/林劼等主编. —北京: 北京大学出版社, 2018. 9  
(高等学校计算机技术类课程规划教材)

ISBN 978-7-301-29776-6

I. ①数… II. ①林… III. ①数据结构—高等学校—教材 ②算法分析—高等学校—教材  
IV. ①TP311.12

中国版本图书馆 CIP 数据核字 (2018) 第 179420 号

|       |  |
|-------|--|
| 书 名   | 数据结构与算法<br>SHUJU JIEGOU YU SUANFA                                |
| 著作责任者 | 林 劼 刘 震 陈端兵 戴 波 主编   |
| 策划编辑  | 孙 晔  |
| 责任编辑  | 巩佳佳  |
| 标准书号  | ISBN 978-7-301-29776-6   |
| 出版发行  | 北京大学出版社  |
| 地 址   | 北京市海淀区成府路 205 号 100871   |
| 网 址   | <a href="http://www.pup.cn">http://www.pup.cn</a> 新浪微博: @北京大学出版社 |
| 电子信箱  | zyjy@pup.cn  |
| 电 话   | 邮购部 62752015 发行部 62750672 编辑部 62754934                           |
| 印 刷 者 | 天津中印联印务有限公司  |
| 经 销 者 | 新华书店   |
|       | 787 毫米×1092 毫米 16 开本 19 印张 471 千字                                |
|       | 2018 年 9 月第 1 版 2018 年 9 月第 1 次印刷                                |
| 定 价   | 48.00 元  |

---

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究

举报电话: 010-62752024 电子信箱: [fd@pup.pku.edu.cn](mailto:fd@pup.pku.edu.cn)

图书如有印装质量问题, 请与出版部联系, 电话: 010-62756370

# 高等学校计算机技术类课程规划教材

## 编委会

---

主任：蒋宗礼 马殿富

副主任：李文新 孙 晔

委员(按姓氏拼音排序)：

蔡之华 陈文智 陈志刚 杜小勇

胡学钢 李凤霞 李仁发 林 菲

刘宏伟 马 啸 秦磊华 石 冰

谭国真 陶先平 王 泉 魏晓辉

吴黎兵 肖鸣宇 俞 勇 袁景凌

翟玉庆 张春元 张国印 张星明



# 前 言

## 一、编写背景

人工智能、大数据已成为现在计算机与信息技术研究领域的重要研究内容,近年来发展迅猛,大量新技术和新兴企业如雨后春笋般孕育而生。许多学生看到一些公司在招聘时要求的编程语言各不相同,便错误地认为学习计算机知识就是学习各种编程语言,或认为学习计算机知识就是学习最新的编程语言、技术和标准。其实大家都被这些公司误导了,编程语言虽然应该学,但是学习计算机核心理论更重要。因为计算机编程语言和开发平台的更新交替很快,但万变不离其宗的是算法设计和数据结构等基础知识。在用计算机解决实际问题的过程中,至关重要的两大环节是实际问题的计算机算法抽象设计和数据结构设计,因此,“算法设计与分析”和“数据结构”便成为计算机相关专业的两门基础核心课程。我们可以生动地把这些基础课程比拟为“内功”,把新的编程语言、技术和标准比拟为“外功”。如果一味地赶时髦,结果可能是只懂得招式,没有功力,这样是不可能成为高手的。

市面上多数教材是将“算法设计与分析”和“数据结构”分开的,主要是因为这两门课程的教学重点有所不同。但实际上,这两门课程的知识共同组成了计算机解决实际问题的主要内容,这两门课程的知识是相互依存、相互衔接、不可分割的。用计算机求解问题时,需要将问题以计算机的方法进行建模和描述,这就是算法设计的过程。算法设计又与数据结构密切相关,不同的算法需要不同的数据结构予以支撑,而不同的数据结构又对算法设计起到一定的制约作用。瑞士著名的计算机科学家、Pascal 程序设计语言之父、结构化程序设计首创者、1984 年图灵奖获得者 N. Wirth 于 1976 年提出了算法 + 数据结构 = 程序 (Algorithms + Data Structures = Programs) 这个公式。在这个著名的公式中,“+”生动地表达了算法和数据结构的相互作用,是程序设计的精髓;“=”形象地刻画了算法和数据结构是构成计算机程序设计的两个关键要素。所以,本书在编写时,我们对“算法设计与分析”和“数据结构”的知识点进行了进一步的梳理,将这两方面知识有机地融入项目驱动的整体化理论实践体系中,让学习者能够真正了解完整的软件设计过程,领略“算法设计与分析”和“数据结构”在整个过程中的相互作用和知识连贯性。让学习者用更少的时间学习到完整的知识和方法,是我们将这两方面知识合编的初衷。

目前,针对算法与数据结构的教学,多数学校仍然用大量课时进行基础理论教学,少量课时进行上机训练。这种教学方式存在的问题是:对于初学者而言,没有上机实践的基础理论学习过于抽象,只能死记硬背,达不到学以致用效果;等到了上机实验课时,学生常常遇到一点点问题就被难倒,甚至无法完成一个简单的上机编程实验,很容易失去学习兴趣。有的学校试点将所有课程都安排在机房,目的是让学生有更多进行编程训练的机会。但实际情况是,有的教师只是换了教室,教学方法却没有做相应的改变或调整,上述问题同样存在;也有的教师发现学生编程时会出现许多问题,一节课下来,什么也没有讲,结果学生的编程操作没有得到很好的指导,而理论课的时间也没有了。翻转课堂教学法可以很好地解决这些问题。在翻转课堂教学中,学生可以先预习课程中简单的基础理论。为了照顾部分自学能力比较弱的学生,任课教师可以将提前录制好的基础理论讲授视频提供给学生,配合 PPT 动画和预习检测题,让学生每周用 2 个学时的时间自学基础理论。学生可以自由选择最适合自己的资料进行自学,并通过完成简单的预习检测题(配答案)来检测自己的预习效果。课堂上,教师可以先通过几分钟的作业检测来检查学生的预习效果,并根据检测结果,灵活调整讲授内容。如果学生的预习效果比较理想,就直接给出应用型的问题,引导学生继续思考,并通过编程解决问题;如果学生的预习效果不理想,则先带领学生进行基础理论复习,可以只针对难点进行细致讲解,然后再引导学生编程解决问题,以巩固所学知识,并进一步培养学生学以致用能力。

电子科技大学计算机科学与工程学院通过三年的教学改革实践,对“数据结构”与“算法设计与分析”两门课程的知识结构进行了重新梳理,结合 MOOC (Massive Open Online Courses,大型开放式网络课程,一般称为“慕课”)教学与翻转课堂教学方法,形成了项目驱动型教学方式,达到了很好的教学效果,并通过对新教学方法的凝练,构成了本书的结构和相关文字内容。

## 二、本书特色

(1)如前所述,其他很多教材多是将“数据结构”和“算法设计与分析”分开讲授,学习者往往无法明白“数据结构”和“算法设计与分析”在解决实际问题中的衔接性和关联性,学习者学习后也无法在一个案例中融会贯通地运用“数据结构”和算法设计思想。本书将“数据结构”和“算法设计与分析”的知识体系重新归纳融合,形成了与其他教材不同的章节内容。可以看到,本书没有其他“数据结构”教材中的“排序查询”章节,这正是通过归纳分析,本书将很多运用了分治算法设计过程的排序查询算法更为合理地划分到了“分治递归”这一章中。通过这样的知识结构安排,学生可以了解到分治递归算法设计思想在查询排序算法设计中的作用。本书还将属于线性数据结构的哈希查询等知识放到了第 2 章“线性结构”中,这样,学生可以对数据结构和对应数据结构上算法设计的本质有更好的理解。同样,在

本书第5章中,我们将图与贪心算法的知识也进行了有机融合。

(2)其他很多教材多是以知识点讲解为主,适合传统的以教师讲授为主的教学。本书以项目驱动的方式编写,每一章都围绕具体案例展开,通过案例引出与案例相关的基础理论知识,学生在学习相关理论后,便可以与教师共同参与案例的解决过程中,从而实现“数据结构”和“算法设计与分析”的连贯性学习。本书第2章、第5章和第6章还提供本章相关知识的其他应用案例,可以作为课堂上教师辅导的资源,更适合翻转课堂教学方式。这样做使得学生能从整体上对问题进行分析,深刻掌握算法和数据结构的综合运用方法,也可以使教学过程更为生动和具有吸引力。如果不采用翻转课堂教学法,则“基础知识”可以安排在课堂上讲解,“项目实战”可以作为理论知识的应用,或在课堂上进行讲解,或作为课后练习作业。

(3)本书可以作为MOOC学习者的进修或补充学习材料。

(4)针对计算机专业(或非计算机专业)的初学者,本书根据学习内容和学习进度,提供完整的问题分析、理论讲解的视频,学习者可以跟着视频进行学习。

### 三、教学安排

本书知识点主要包括:线性数据结构、栈与队列数据结构、树数据结构、图数据结构、分治递归算法设计、贪心算法设计、动态规划算法设计、算法复杂度分析、排序与查找等。具体学时安排可参考下表。

本书各章学时安排

| 各章内容       | 讲授学时 | 实践学时 |
|------------|------|------|
| 第1章 绪论     | 4    | 0    |
| 第2章 线性结构   | 8    | 10   |
| 第3章 递归与分治  | 4    | 4    |
| 第4章 树      | 6    | 7    |
| 第5章 图与贪心算法 | 6    | 7    |
| 第6章 动态规划   | 4    | 4    |
| 合计         | 32   | 32   |

## 四、教学方法

本书可以采用两种教学方法。

### 1. 传统教学法

第1章的内容及对应PPT可以作为课堂教学内容,章后习题可以作为学生的课后作业。

第2章到第6章,每一章中的“项目指引”与“基础知识”可以作为课堂教学内容;“项目实战”可以作为课堂讨论内容,其他案例可以作为学生课后实践内容。

## 2. 翻转课堂教学法

第1章的内容及对应PPT可以作为课堂教学内容,章后习题可以作为学生的课后作业。

第2章到第6章,每一章中的“项目指引”与“基础知识”(有配套的视频)可以作为学生每周的课前预习资料;“项目实战”可以作为课堂上教师引导学生完成的课堂作业,完成后可以通过学生展示的方式,共同解答并总结。课堂上,教师和学生在进行项目实战时,可以首先选择一个案例进行分析讲解,等学生对相关知识掌握比较好之后,便可以其他案例进行翻转课堂教学,供学生在有教师辅导条件下进行项目实战练习。不同基础的学生完成项目数目可以有所不同。

## 五、致谢

本书由刘震、戴波、陈端兵和林劼四位教师合作编写。刘震编写了第1章和第3章,戴波编写了第2章,陈端兵编写了第4章,林劼编写了第5章和第6章。由于编者水平有限,书中难免有错误与不足之处,欢迎广大读者批评指正,我们会在本书重印或再版时做出必要修正。北京大学出版社的巩佳佳编辑为本书的出版做了大量工作,笔者在此表示由衷感谢。本书在编写过程中,参阅了一些相关资料,在此,向这些资料的相关作者表示感谢!

## 六、参考书目

- [1] 吴跃. 数据结构与算法[M]. 北京:机械工业出版社,2010.
- [2] 李春葆. 数据结构教程:C#语言描述[M]. 北京:清华大学出版社,2013.
- [3] 王晓东,等. 算法设计与分析[M]. 北京:清华大学出版社,2014.
- [4] Cormen,等. 算法导论:第3版[M]. 殷建平,等译. 北京:机械工业出版社,2013.





# 目 录

|                            |    |
|----------------------------|----|
| 第1章 绪论 .....               | 1  |
| 1.1 数据结构与算法的发展简史 .....     | 1  |
| 1.2 利用计算机求解问题的一般过程 .....   | 2  |
| 1.3 数据结构的基本概念和术语 .....     | 5  |
| 1.3.1 数据的基本特性 .....        | 5  |
| 1.3.2 数据的逻辑结构 .....        | 6  |
| 1.3.3 数据的存储结构 .....        | 7  |
| 1.3.4 数据结构的定义 .....        | 8  |
| 1.3.5 数据类型 .....           | 8  |
| 1.3.6 抽象数据类型的含义与表示形式 ..... | 9  |
| 1.4 算法的概念 .....            | 9  |
| 1.5 算法的复杂度分析 .....         | 11 |
| 第2章 线性结构 .....             | 15 |
| 2.1 项目指引 .....             | 15 |
| 项目1 电话号码本 .....            | 15 |
| 项目2 迷宫寻路 .....             | 15 |
| 项目3 自助交易平台 .....           | 15 |
| 项目4 电话号码本的快速查找 .....       | 16 |
| 2.2 基础知识 .....             | 16 |
| 2.2.1 线性表 .....            | 16 |
| 2.2.2 特殊线性表 .....          | 39 |
| 2.2.3 线性表的查找算法 .....       | 55 |
| 2.2.4 线性表的排序算法 .....       | 67 |
| 2.3 项目实战(任务解答) .....       | 77 |
| 项目1 电话号码本 .....            | 77 |
| 项目2 迷宫寻路 .....             | 82 |
| 项目3 自助交易平台 .....           | 86 |
| 项目4 电话号码本的快速查找 .....       | 87 |
| 2.4 线性结构的其他应用 .....        | 91 |
| 2.4.1 线性表的简单应用 .....       | 91 |
| 2.4.2 栈的简单应用 .....         | 95 |

|                     |     |
|---------------------|-----|
| <b>第3章 递归与分治</b>    | 107 |
| 3.1 项目指引            | 107 |
| 项目1 查找和排序问题         | 107 |
| 项目2 汉诺塔问题           | 109 |
| 项目3 大整数乘法问题         | 110 |
| 项目4 分形问题            | 110 |
| 3.2 基础知识            | 113 |
| 3.2.1 递归的概念         | 113 |
| 3.2.2 递归与迭代的区别      | 114 |
| 3.2.3 递归与栈的关系       | 115 |
| 3.2.4 分治的原理         | 115 |
| 3.3 项目实战(任务解答)      | 118 |
| 项目1 查找和排序问题         | 119 |
| 项目2 汉诺塔问题           | 126 |
| 项目3 大整数乘法问题         | 129 |
| 项目4 分形问题            | 132 |
| <b>第4章 树</b>        | 136 |
| 4.1 项目指引            | 136 |
| 项目1 查找与排序问题         | 136 |
| 项目2 压缩编码问题          | 136 |
| 项目3 频繁模式挖掘问题        | 136 |
| 4.2 基础知识            | 137 |
| 4.2.1 二叉树、树及森林的基本概念 | 137 |
| 4.2.2 二叉树的基本性质      | 140 |
| 4.2.3 二叉树、树及森林的存储结构 | 142 |
| 4.2.4 二叉树、树及森林的基本操作 | 148 |
| 4.3 项目实战(任务解答)      | 157 |
| 项目1 查找与排序问题         | 157 |
| 项目2 压缩编码问题          | 186 |
| 项目3 频繁模式挖掘问题        | 191 |
| <b>第5章 图与贪心算法</b>   | 203 |
| 5.1 项目指引            | 203 |
| 路由协议设计              | 203 |
| 5.2 基础知识            | 204 |
| 5.2.1 图             | 204 |
| 5.2.2 贪心算法          | 218 |
| 5.3 项目实战(任务解答)      | 222 |
| 路由协议设计              | 222 |
| 5.4 更多案例            | 232 |

|       |                          |     |
|-------|--------------------------|-----|
| 5.4.1 | 高速公路建设问题 .....           | 232 |
| 5.4.2 | 项目工作流程规划问题 .....         | 242 |
| 5.4.3 | 其他图应用举例 .....            | 251 |
| 第6章   | 动态规划 .....               | 258 |
| 6.1   | 项目指引 .....               | 258 |
|       | 项目1 实验报告防抄袭小系统设计问题 ..... | 258 |
|       | 项目2 路由协议设计问题 .....       | 258 |
| 6.2   | 基础知识 .....               | 259 |
| 6.2.1 | 动态规划简介 .....             | 259 |
| 6.2.2 | 动态规划算法的基本设计思想 .....      | 260 |
| 6.2.3 | 动态规划要素 .....             | 267 |
| 6.3   | 项目实战(任务解答) .....         | 269 |
|       | 项目1 实验报告防抄袭小系统设计问题 ..... | 270 |
|       | 项目2 路由协议设计问题 .....       | 275 |
| 6.4   | 更多案例 .....               | 279 |
| 6.4.1 | “0-1”背包问题 .....          | 279 |
| 6.4.2 | 装配线调度问题 .....            | 282 |
| 6.4.3 | 权重化的活动安排问题 .....         | 286 |
| 6.5   | 贪心算法与动态规划算法的差异 .....     | 288 |

# 第1章 绪论

## 1.1 数据结构与算法的发展简史

数据结构与算法是计算软科学的基础课程,这部分内容充分体现了计算理论与计算机技术相结合发展的历史沿革,是计算机科学与技术专业学生踏入计算软科学大门的金钥匙。不同于这门课程的前修课程,如“程序设计语言”课程主要偏重于学习计算机编程语言的语法细节,“离散数学”则仍然偏重于理解和掌握较艰深的数学理论,给学生的印象是与计算机技术本身相去甚远;而通过这门课程的学习才开始真正让学生感受与领略到计算软科学的无穷魅力并逐渐形成一定的计算思维。

随着1946年世界上第一台电子计算机ENIAC在美国宾夕法尼亚大学诞生,人们开始利用计算机代替人工处理各种数据。早期数据的存储形式主要是二进制编码,以整数和浮点数这样的“无结构”数据为主。之后10多年时间,随着程序设计语言的诞生和发展,人们开始考虑如何高效地处理一些更复杂、规模更大的数据,这就要求数据必须按照某些特定的格式和规则进行存储。一些相对比较复杂的结构化数据(例如,数组、栈、散列表和二叉树等)逐渐出现在程序设计语言中,大大促进了计算机处理数据的效率和能力。

“数据结构”的概念最早由英国计算机科学家C. A. R. Hoare和瑞士计算机科学家N. Wirth在1966年提出。大量关于程序设计理论的研究表明:为了系统而科学地构造大型复杂的程序,必须对这些程序中所包含的数据结构进行深入的研究。1968年,美国科学家D. E. Knuth在他的名著《计算机程序设计艺术》(第1卷:基本算法,第2章:信息结构)中首次系统地研究并整理了当时经常使用的主要数据结构与相关的算法,为数据结构课程的开设提供了丰富的素材(他本人也因此书的成就,在1974年获得世界范围内计算机科学领域最高科学成就奖“图灵奖”)。

有很多人认为20世纪60年代是数据结构发展成熟的年代,但是美国科学家J. Hopcroft认为“数据结构与算法”这个学科是从20世纪70年代才开始兴起,并逐渐成为计算机科学的重要组成部分的。由于Hopcroft和另一位美国科学家R. E. Tarjan在这方面的突出贡献,他们共同获得了1986年的“图灵奖”,这肯定了他们在算法与数据结构设计和分析的基础成就。这一奖励说明“数据结构与算法”在当时才真正发展到比较成熟的程度,并得到了人们的公认,同时也说明它在计算机科学中的重要地位。自20世纪70年代起,“数据结构与算法”在西方国家的大学中,被普遍列为计算机本科的必修专业核心课程。“数据结构与算



视频讲解

法”课程的重要性不言而喻。目前,它已被列为国内很多普通高校的计算机相关专业研究生入学考试的必考科目,也是软件人员水平考试的重要内容;要在程序设计竞赛(如 ACM 程序设计大赛)中取得优秀的成绩,也需要以这门课程的学习作为基础。

那么,这门课程应该怎么入门呢?我们先来看一下 N. Wirth 在 1976 年提出的一个公式  

$$\text{程序} = \text{数据结构} + \text{算法}。$$

在这个著名的经典公式中,“+”生动地表达出算法和数据结构的相互作用,它们是程序设计的精髓;“=”言简意赅地刻画出算法和数据结构是构成计算机程序的两个关键要素。计算机程序是使用计算机求解问题的若干逻辑步骤的集合,问题的有效求解涉及算法和数据结构这两个不可分割的部分。简单地理解,数据结构就是对所要求解问题中的数据的存储和表示,而算法是求解问题的过程描述。那么,有哪些常用的数据结构?又有哪些经典的算法策略?如何设计数据结构?如何设计算法?这些问题都将是本书要讨论的主要内容。接下来,让我们通过本书的学习去领略各种数据结构的功用和特点,去感受各种算法策略的精妙和乐趣吧!

## 1.2 利用计算机求解问题的一般过程

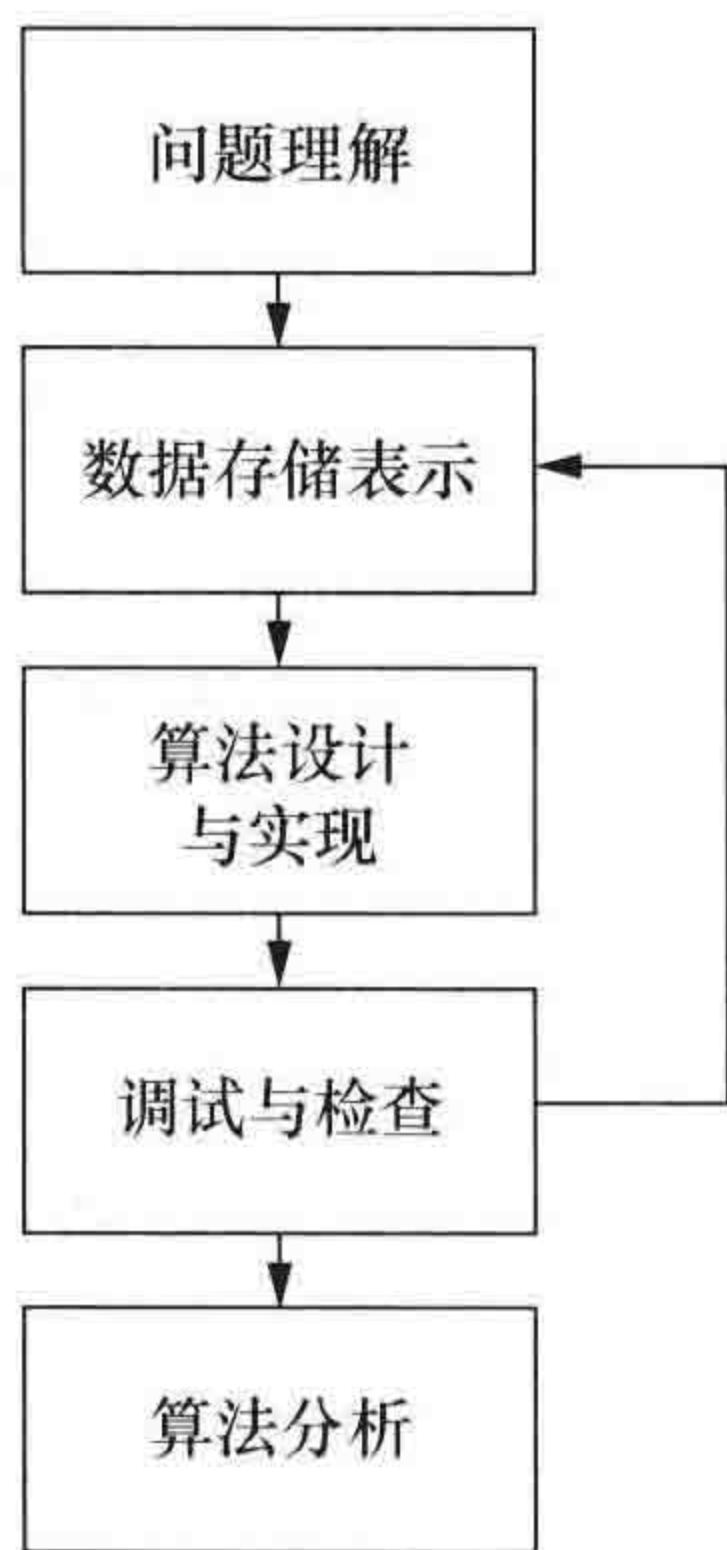


图 1.1 计算机求解问题的一般步骤

我们需要特别强调的是,学习本门课程的过程是一个系统培养和锻炼计算思维的过程。为了说明这一点,我们先从一般的角度来看用计算机求解问题时通常需要怎样的一个过程。回顾一下,当刚学习和掌握一门编程语言并希望借助编程来处理 and 解决一个新的问题时,我们通常会按照图 1.1 中所示的流程来开展我们的工作。下面我们通过一个具体的例子来说明这个流程,这是一道 ACM 程序设计大赛的题目。

现在有一种新停车场称为停车楼。停车楼的原理是这样的,你开车进入停车楼的第一层入口电梯,然后电梯和对应楼层的传送带可以将你的车输送到某一楼层的一个空的停车位;当你回来取车时,电梯和传送带又可以将你的车送回到停车楼的第一层入口电梯。

停车楼的布局也比较简单,有一个中央电梯负责在楼层间输送汽车,在每一层有一个大型的循环传送带系统。传送带可以沿顺时针或者逆时针方向移动。当电梯到达了某一楼层,它就成为传送带的一部分,这样汽车就可以被传送带移动到某个空的停车位了。

在每一天的下班时间,停车楼通常停满了车,而且会有很多人过来取走他们的车。这里提供的服务按照先来先服务的顺序。也就是说,根据用户排队的顺序,电梯先到第一个用户的车所在楼层,该楼层的传送带将用户的车移动到电梯口,然后电梯再回到第一层的入口;第一个用户取走车后,轮到下一个用户取车。我们的问题是想知道所有车都被取走需要花费的时间。已知电梯上行一个楼层或者下行一个楼层的时间是 10 秒钟,传送带将车沿任意方向(顺势针或者逆时针)移动一个车位的时间是 5 秒钟。

输入:

第一行是一个正数,这个数代表测试用例的个数,至多不超过 100,在这个数值之后是每个测试用例。

(1)这一行有两个整数: $h$  和  $l$ ,其中, $1 \leq h \leq 50, 2 \leq l \leq 50$ 。 $h$  和  $l$  分别表示停车楼的楼层总数和传送带的长度。

(2)用  $h$  行并且每行有  $l$  个整数表示所有车的初始泊车位置。第  $i$  行的第  $j$  个数表示第  $i$  层的第  $j$  个停车位置。对应的数  $r$  表示该停车位的车应该是第  $r$  个可以被取走的车,如果  $r$  的值为  $-1$ ,则表示该停车位尚未停泊车。电梯入口是在第一层;电梯初始时是空的,并且在第一层的第一个停车位置。停车楼至少停有一辆车。

输出:

对每个测试用例,用一个数(秒)对应一行,表示所有车被取走的时间。

示例输入:

```
2
1 5
-1 2 1 -1 3
3 6
-1 5 6 -1 -1 3
-1 -1 7 -1 2 9
-1 10 4 1 8 -1
```

示例输出:

```
25
320
```

针对这个问题,我们通过下面五个步骤来分析和解决。

第一步,我们需要分析和理解题意。根据问题的描述,我们要计算所有车被取走的时间,需要先分别计算每辆车被取走的时间。而计算每辆车被取走的时间,关键是要定位每辆车所停放的楼层和传送带的位置,然后根据停车位置距离电梯口的距离,判断选择让传送带进行顺时针移动或者逆时针移动。下面,我们先看看示例输入的第一个测试用例的计算过程(如图 1.2 所示)。

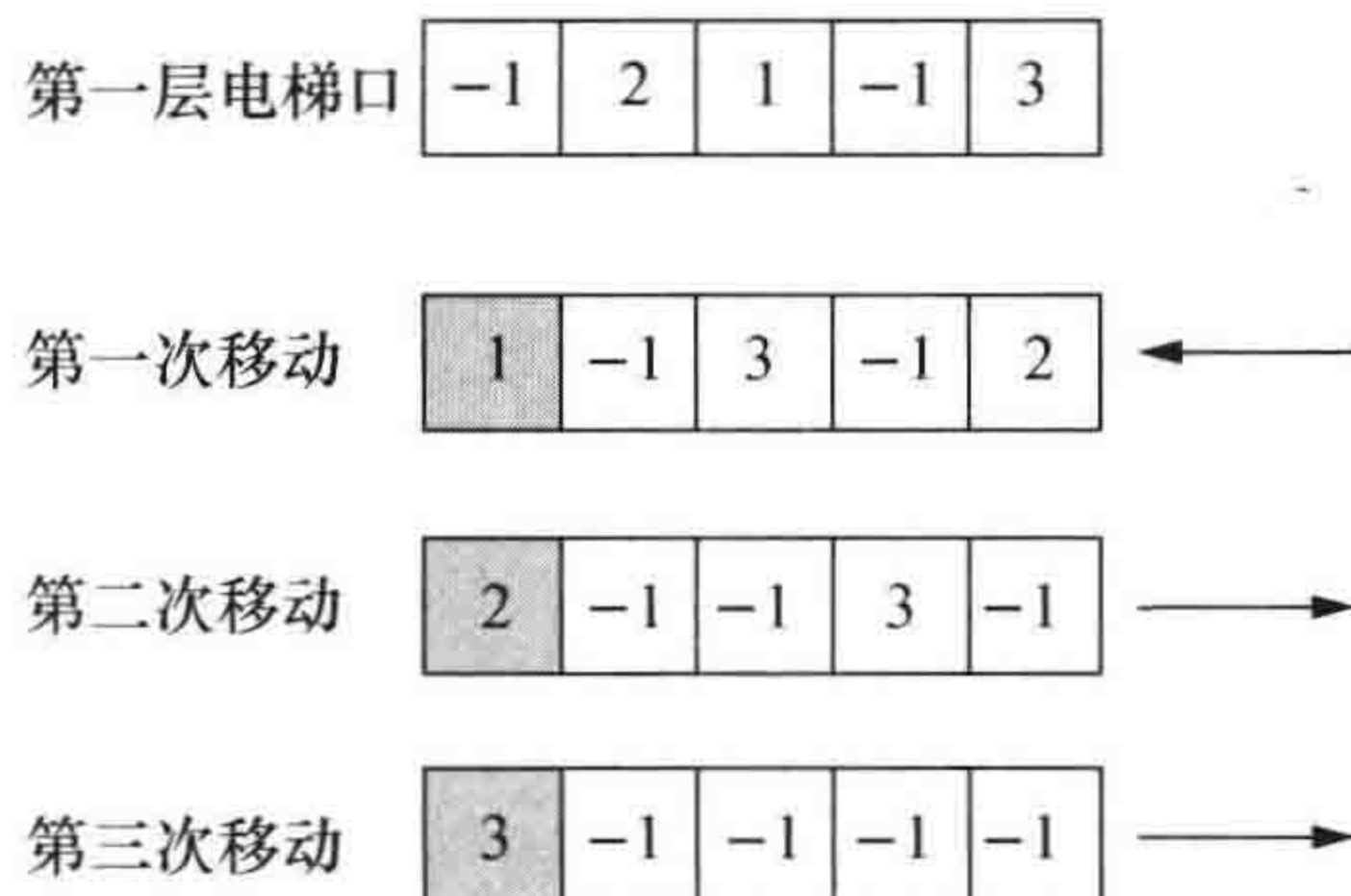


图 1.2 传动带的移动过程示例

由于三辆车都停放在了第一层,因此只需要传送带进行三次移动,而不需要电梯的上下

移动。1号车离电梯口只有两个车位的距离,因此传送带只需要左移两个车位,1号车就可以被取出,用时10秒钟。完成第一次移动后,根据2号车的位置,如果将其向左移动,到电梯口的距离较远;但如果让传送带循环右移,则只需要移动一个车位,2号车就可以到电梯口,因此,选择让传送带右移,将2号车取出用时5秒。2号车被取走后,可以通过让传动带循环右移两个车位,使3号车最快到达电梯口,需要10秒。所以,对于第一个测试用例,总的用时应该是25秒。第二个测试用例与第一个测试用例类似,但还需要统计电梯上下楼层的时间,这里不再赘述。通过以上分析,我们已经完全理解了这个问题的要求。

第二步,针对问题的输入,我们需要考虑如何表示和存储数据。输入主要包含三项内容,分别是用例个数、楼层数和传送带长度,以及所有车辆的停车位置。对于用例个数、楼层数和传送带长度,可以简单地用三个 Integer 类型的变量 Num\_TestCase, Num\_Floor 和 Len\_Belt 存储;而对于所有车的停放位置,我们可以用一个 Integer 类型的二维数组 Park\_Pos[floor][belt] 来表示,例如, Park\_Pos[3][2] = 5 表示第5号车停放在第3层的2号车位。

第三步,在选择了适当的数据表示和存储方法的基础上,我们需要选择某种计算策略,设计一定的步骤操作数据,并编程计算得到问题要求的结果。对每个测试用例,我们可以按照以下步骤进行编程实现:

- (1) 在 Park\_Pos 数组上查找得到第  $k$  辆车的数组下标,即楼层位置  $i$  和车位位置  $j$ 。
- (2) 计算从当前楼层到第  $k$  辆车所在楼层的电梯移动时间。
- (3) 沿传送带左移方向和右移方向分别计算当前车位位置与电梯口的车位距离。
- (4) 如果左移车位距离电梯口比右移车位距离电梯口更近,则对 Park\_Pos 数组的第  $i$  行数据做循环左移,直到第  $k$  号车被移动到电梯口为止,计算移动时间并累加到获取该车辆的总时间中;否则,对 Park\_Pos 数组的第  $i$  行数据做循环右移,直到第  $k$  号车被移动到电梯口为止,并做相应计算。然后,将 Park\_Pos 数组的第  $i$  行第一个元素的值置为  $-1$ 。
- (5) 计算从当前楼层到一楼出口的电梯下行时间,并累加到获取该车辆的总时间中。
- (6) 回到步骤(1),计算下一辆车的取出时间,直到统计完所有车辆为止。

第四步,如果算法第三步得到的结果与我们期望的结果不一致,我们还会回到第二步和第三步,对数据表示和计算过程进行检查和修改,直到最终得到我们想要的正确结果。这也是我们在编程实现的过程中经常会反复遇到的状况。

第五步,虽然我们通过编程实现得到了正确结果,但为了提高程序的运行速度,降低数据占用的计算机存储空间,我们可能还会进一步地优化和改进数据表示的方式和计算的方法。例如,针对上述这个问题,对于所有车辆的停放位置数据,我们除了可以采用二维数组进行存储以外,也可以使用结构数组进行存储。通常我们也会考虑通过改进查找第  $k$  辆车位置的算法来提高算法效率。

在以上整个问题求解的过程中,虽然我们并没有学习过数据结构与算法这门课程,但我们已经不经意地应用了数据结构与算法这门课程中将要介绍的一些知识和方法。其中,第二步涉及数据结构的知识,第三步涉及算法的知识,第五步则涉及算法评价与分析的知识。在这门课程接下来的学习中,我们将分别针对这些内容进行系统的介绍,让大家能够更清晰地认识和理解“数据结构与算法”这门课程的内涵和外延。



## 1.3 数据结构的基本概念和术语

### 1.3.1 数据的基本特性

要理解什么是数据结构,首先需要知道什么是数据。对于这个问题,可能每个人都有自己的理解。维基百科给出的定义是:数据是关于事件之一组离散且客观的事实描述,是构成信息和知识的原始材料。这个定义比较抽象,而百度百科则给出了更具体的定义:科学实验、检验、统计等所获得的和用于科学研究、技术设计、查证、决策等的数值都是常见的数据。

本书认为,数据是指存储在某种介质上能够识别的物理符号,是信息的载体,这些符号可以是数、字符或者其他。下面我们来看一个例子。

假设我们有某大学计算机系全部入学新生的基本信息数据,其中新生陈小丽的基本信息如图 1.3 所示。图中的身高、体重、性别、入学总分和学号都是与陈小丽相关的个人数据。注意,虽然数据经常以数字的形式呈现在我们面前,但不能简单地将数据理解成数字,因为数据还有很多其他的形式,比如声音数据、图形数据和视频数据等。

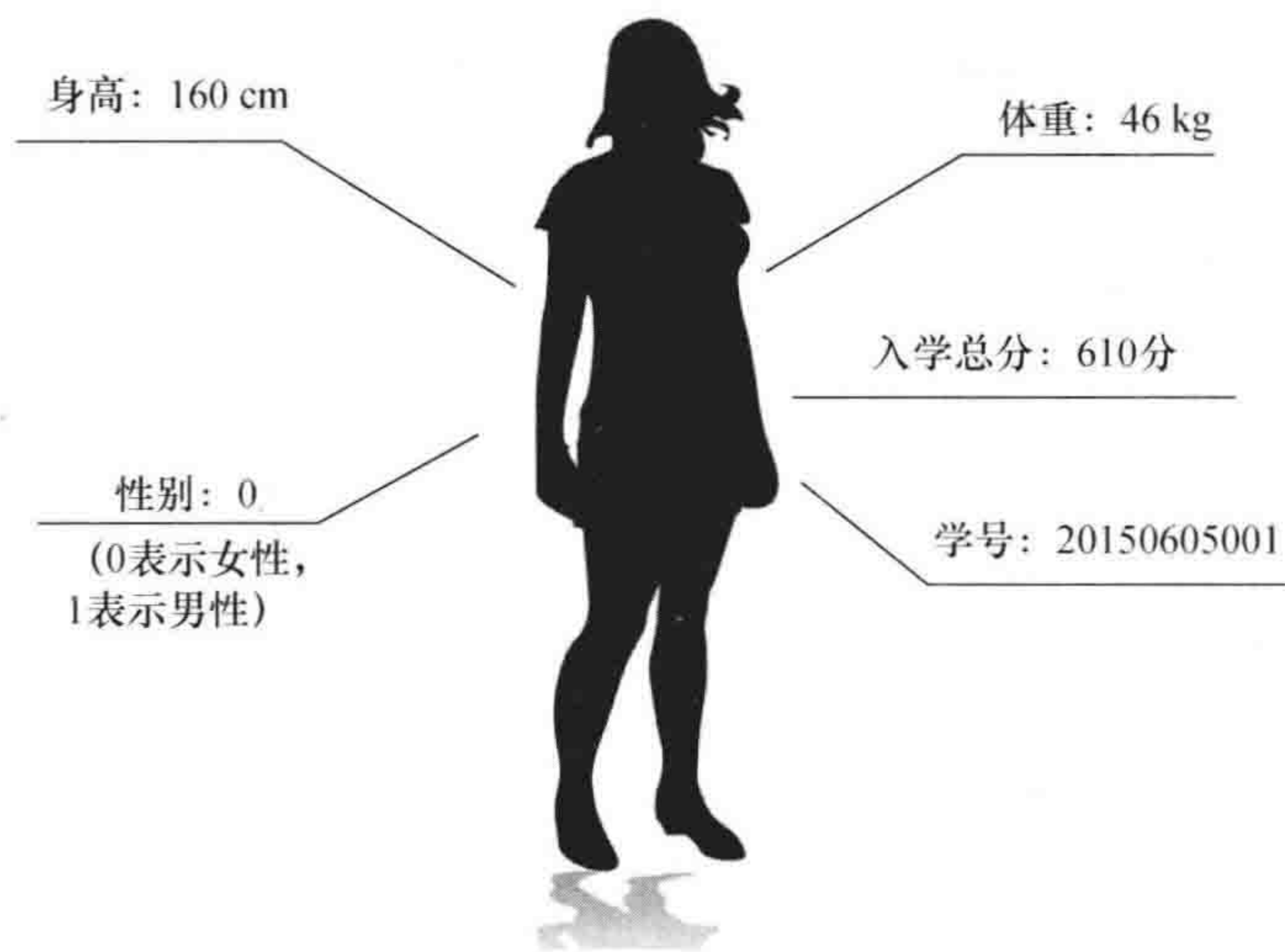


图 1.3 大学入学新生陈小丽的基本数据

那么,数据为什么需要结构呢?我们来看一下,如果把陈小丽的个人数据堆放在一起,形成一串很长的数字而不给出一定的结构约束它[如图 1.4(a)所示],那么这串数字将是无法让人理解的。但如果我们按照身高、性别、体重、入学总分和学号的顺序对这串数字进行结构划分[如图 1.4(b)所示],这串数字的意义就很容易理解了。

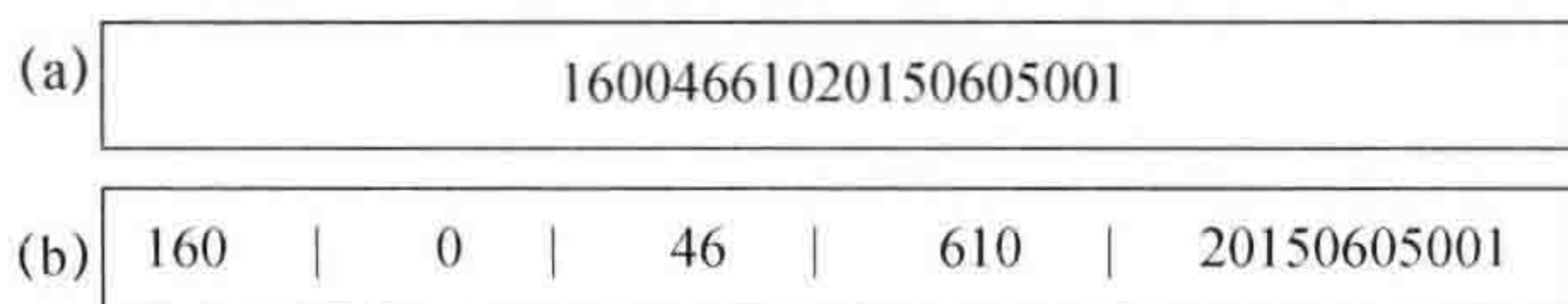


图 1.4 有结构的数据与没有结构的数据



因此,这里我们得到了一个重要结论:一堆杂乱无章的数据是无法理解和使用的;只有对数据定义出适当的结构,才能方便我们理解和使用。同时,数据需要基本的构成部件,就像物体是由分子构成的,分子是由原子构成的一样。数据是由数据元素构成的,而数据元素是由数据项构成的。例如,计算机系所有入学新生的数据是由许多像陈小丽这样的单个数据元素构成的;而陈小丽个人信息的数据元素又是由身高、体重、性别等数据项构成的。我们将数据元素称为构成数据的基本单位,而数据项是构成数据的最小单位。它们之间的关系如图 1.5 所示。

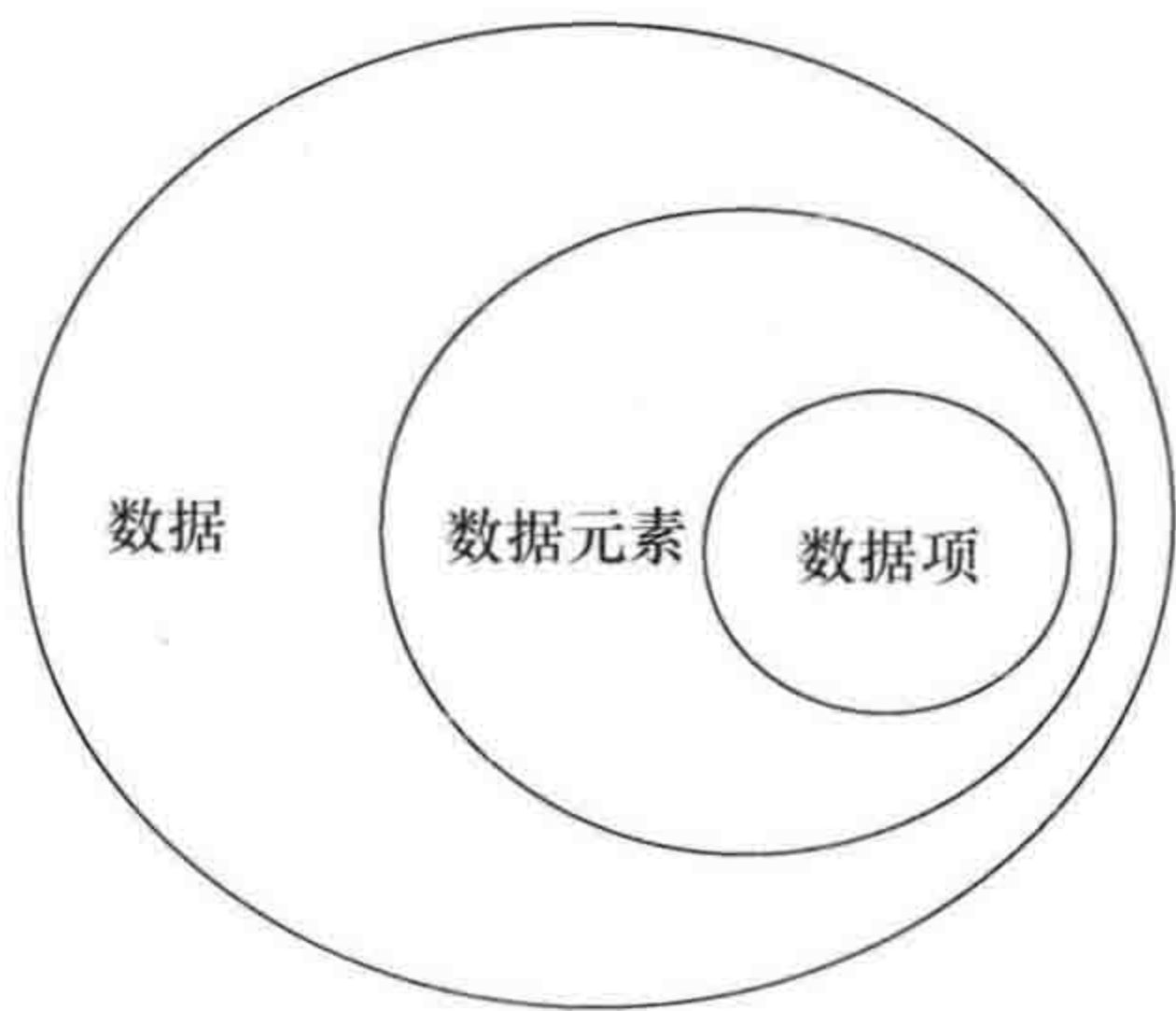


图 1.5 数据的基本构成及其关系

### 1.3.2 数据的逻辑结构

假设我们需要编制一个项目管理的程序,来管理一个公司所有项目人员的各项事务。那么,这个公司的管理架构可以是这样的:总经理管理所有项目的项目经理,每个项目小组由一位项目经理和若干个开发人员组成,其中开发人员由项目经理直接负责管理。该公司项目人员的管理组织结构如图 1.6 所示。

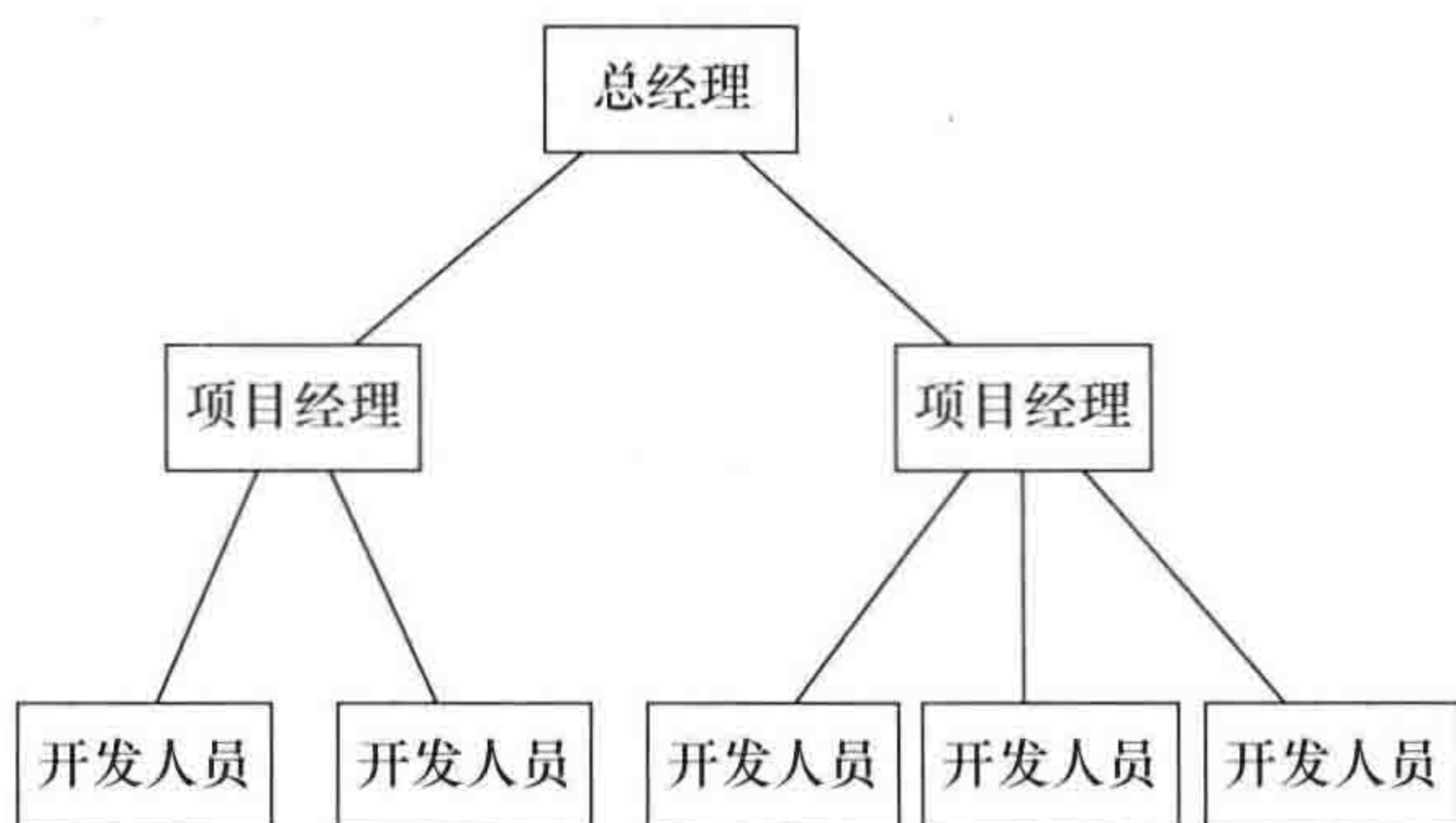


图 1.6 某公司项目人员的管理组织结构

在这样的组织结构中,每个数据元素(项目人员)都存在领导或被领导的关系。因此,我们又可以得到一个重要结论,数据元素之间通常是存在关系的。我们将数据元素之间的关系系统称为逻辑关系或者逻辑结构,那么数据元素之间可能存在哪些关系呢?

在现实生活中,我们大概能分析和整理出四大类常见的数据关系。第一种是集合关系,例如,一个班级里身高超过 180cm 的同学可以构成一个集合,班上爱好画画的同学也可以构

