

广西高校教学名师专项经费资助项目

广西特色专业及课程一体化建设项目经费资助项目

广西高等教育教学改革工程重点项目经费资助项目

数据结构与算法

——理论与实践

| 唐培和 徐奕奕 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

广西高校教学名师专项经费资助项目

广西特色专业及课程一体化建设项目经费资助项目

广西高等教育教学改革工程重点项目经费资助项目

数据结构与算法

——理论与实践

唐培和 徐奕奕 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以数据的逻辑结构为线索,介绍数据结构及其建立在数据结构之上的操作和算法,内容涉及数据结构与算法的基本概念、顺序存储结构的线性表、链式存储结构的线性表、串、数组、特殊矩阵与广义表、树与二叉树、图及其应用、查找与搜索引擎、排序等。

本书既强调“算法”与“数据结构”之间紧密的依赖关系,也注重算法描述的简洁与干练;既有完整的理论体系,也有很好的应用案例。为了增强可理解性,本书案例与生活实践相联系。

本书以知识单元为基本构件,既可拆卸也可重组,内容丰富,表述详尽,可读性强,适合不同类型的本科院校按照不同的培养规格组织教学。

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。

图书在版编目(CIP)数据

数据结构与算法:理论与实践 / 唐培和, 徐奕奕编著. —北京: 电子工业出版社, 2015.1

ISBN 978-7-121-24467-4

I. ① 数… II. ① 唐… ② 徐… III. ① 数据结构 ② 算法分析 IV. ① TP311.12

中国版本图书馆 CIP 数据核字 (2014) 第 230398 号

策划编辑: 章海涛

责任编辑: 章海涛

特约编辑: 何 雄

印 刷: 涿州市京南印刷厂

装 订: 涿州市京南印刷厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 24.25 字数: 700 千字

版 次: 2015 年 1 月第 1 版

印 次: 2015 年 1 月第 1 次印刷

定 价: 52.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

《数据结构与算法》作为一门独立的课程始于 1968 年。那时在美国一些大学中，虽然把《数据结构》规定为一门课程，但对其范围仍没有明确界定。当时，“数据结构”几乎与图论、表、树的理论为同义语。由于数据必须在计算机中进行处理，因此不但应考虑数据本身的数学性质，而且必须考虑数据的存储结构。随着数据库系统的不断发展，在数据结构课程中又增加了文件管理的内容。20 世纪 60 年代末到 70 年代初出现了大型程序，软件也相对独立，结构化程序设计成为程序设计方法学的主要内容，人们开始越来越重视数据结构，认为程序设计的实质是对确定的问题选择一种好的数据结构，加上设计一种好的算法。从 20 世纪 70 年代中期到 80 年代初，各种版本的数据结构著作就相继出现。

也就是说，在计算机专业教育中，刚开始并没有《数据结构与算法》这样的课程。后来人们意识到，在系统软件与应用软件设计中有不少“数据结构”和“算法”是常用的、普适的，把它们抽象、提炼出来，就形成了这门课程。这种抽象与提炼不仅必要，也十分有意义。就像文学作品一样，应该“源于生活，高于生活，给人们以美的熏陶与享受”。

众所周知，算法侧重于解决问题的思想与方法，程序强调的是计算机上的实现。因此，算法比程序更宏观、更抽象、更简练、更易懂，讲解算法比讲解程序更容易让学术掌握算法的本质与核心。如果用一种具体的语言来描述算法，会使学生陷于程序的语法细节而不能自拔，从而影响学生对算法的理解。

正因为算法“源于”程序、算法“高于”程序，所以本书作者不赞成用一种具体语言（如 C/C++ 语言）来描述算法，尽管这种“功利性”的教材市面上很多。

不妨看一个非常简单的例子——输出数组所有元素的值，为保证原汁原味，编码格式都没做调整。形式如下所示：

```
template <class T>
void PrintArray(T list[], int n)
{
    while (n>0){
        cout<<list[n-1]<<" ";
        --n;
    }
}
```

针对此例，如果按本教材的风格写成算法，形式如下：

```
void PrintArray( list[], n)
{
    while ( n > 0 )
    {
        print( list[n-1] );
```

```

        n ← n - 1;
    }
}

```

对比一下，不难看出哪种方式更有利于教学。

从另一个角度来说，如何把算法转换成程序正是该课程重要的教学目标之一。如果学生课后能把教材和课堂上讲解的算法转换成由某一种语言描述的程序并上机实践，就能加深学生对算法的理解以及提升学生的实践能力。

因此，从教学的角度来说，当学生学完某一门具体的计算机语言、学完数据结构与算法、学完程序设计方法学的相关知识后，通过一些综合性的课题或项目来综合这些理论、知识与技能，以达到真正意义上的“融会”与“贯通”。而不是过早地把它们融合在一起，让学生抓不住本质、摸不着头脑。

另外，数据结构与算法与程序设计方法学（如面向对象方法学）没有必然的联系。笔者认为，只有学生真正应用算法设计系统或应用程序时，才应该考虑程序设计方法学方面的问题。否则，算法设计与某种具体的程序设计方法学相结合，就会使算法复杂化，从而影响学生对算法的理解。

基于以上认知，本书事先定义一种简洁的、类 C 语言的算法描述语言，所有算法都用它来描述，然后要求或鼓励学生利用实验或课外条件完成由算法到程序的转换。

本书除了作为课堂教学重要的参考书外，主要是给学生看的。课前看，叫预习；课后看，叫复习。课堂上显然不需要学生看教材，否则教师的讲课就多余了。因此，如何让学生容易读懂教材应该作为衡量教材的一个重要指标。最忌讳的是把教材写成了学术著作，以展示作者的学术水平。从这个角度上来说，本书力求用生活中的一些例子来阐明一些理论和思想（即贴近“生活”），尽量体现通俗易懂，尽量不要让学生“望而生畏”。尽管这样做可能会使本书不那么“阳春白雪”，但只要对教学有利，还是很值得的。

站在学习者的角度来说，学什么都希望了解“前因后果”，即为什么要学这些东西？以及学习这些东西有什么意义（或者有什么用）？作为教师或者教材，应该给出相应的解答，否则学习者是缺乏“本源动力”的。本书在这方面应该说尽力而为了。比如，第 8 章“查找与搜索引擎”，除了讲述为什么要学习“查找”外，用较大篇幅介绍 Google 搜索引擎，相信会激发学习者的兴趣。事实上，Google 是数据结构与算法的集大成者，又是非常成功的典型代表。如果读者具有了较好的程序设计能力，相信学完本课程后，自己都能构造一个简单的、专用的搜索引擎！

全书共 9 章。

第 1 章介绍数据结构与算法的基本概念，并对此进行详尽的解读，如算法设计的准则、算法与程序的对比等。

第 2 章介绍顺序存储结构的线性表，侧重顺序存储结构的特点及其建立在该结构上的操作（算法）。

第 3 章介绍链式存储结构的线性表，强调其与顺序存储结构的差异与互补及其建立在链表结构上的操作，并介绍静态链表及其应用。

第4章介绍串及其基本概念与操作，重点在串的模式匹配。

第5章介绍数组、特殊矩阵和广义表，并引出数据压缩与人工智能等概念。

第6章介绍树与二叉树，重点在二叉树及其应用，如哈夫曼编码、二叉排序树等。

第7章介绍图及其应用，设计的概念较多，重在图的存储结构、遍历及其应用，如最小生成树、最短路径、拓扑排序等。

第8章介绍查找与搜索引擎，在讲解基本的查找方法后，结合 Google 重点讲述搜索引擎的构造思想和方法。

第9章介绍几大类排序方法，如插入排序、选择排序、交换排序、归并排序、基数排序等。

本课程相对来说比较抽象，学生不容易理解和掌握。虽然教材或黑板上的示意图可以帮助人们理解，但用一个节点表示一个数据元素，用一根线段表示数据元素之间的关系，这样画出来的示意图本身就是比较抽象的。要在此基础上理解其存储结构、建立在存储结构上的算法以及算法的执行情况，自然变得非常困难，这就是本课程难以学习和掌握的症结所在。为之，人们开发了各种教学软件，帮助学生学习数据结构与算法。

为此，作者组织人力开发了“数据结构与算法综合集成演示系统”，本系统集成图、链表、矩阵转置、二叉树、排序、数据查找等主要内容，涵盖了《数据结构与算法》课程的绝大部分内容，集成度高，以非常直观、方便的方式演示了图、二叉树等复杂的数据结构元素之间的层次及网状关系。本系统基于相应的数据结构，演示了各种数据结构基础之上的算法，如遍历、最短矩阵、拓扑排序等。本系统还集成了演示动画。

例如，对于网状结构而言，可以利用作图元直接在屏幕上绘制数据结构的网状图，根据网状图可自动转换成机器内部的数据结构(邻接矩阵或邻接表)，绘制的网状结构可以文件的形式存储，实现了基于邻接表与邻接矩阵的各种相关算法，并可查看相关算法和代码。如图1所示。

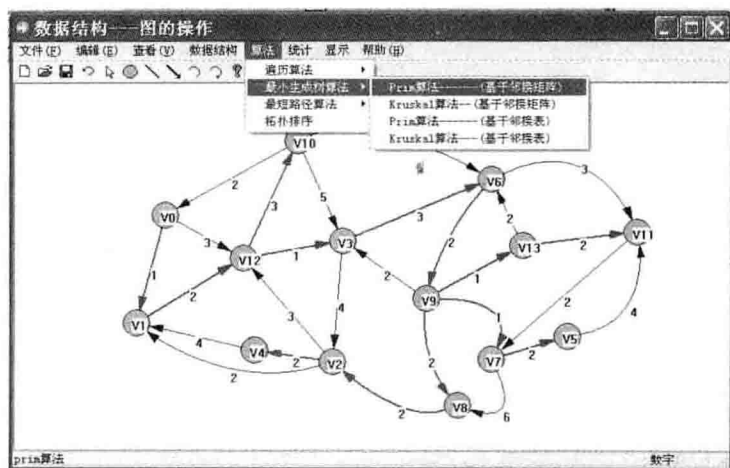


图1

又如，学生对各种排序算法的理解相对容易，但对各种不同算法的效果难以体会。本系统囊括了课程里面几乎所有的排序算法，可自动产生 100~10000000 个数据供测试算法效率用。根据

用户的选择，调用相应的排序算法，可精确记录排序算法所耗费的时间，能显示排序的结果并用柱状图、时间列表来对比排序算法的优劣性。

全书由唐培和教授负责统筹、安排、协调、统稿、审核等，其中第4章、第5章以及每章习题由徐奕奕副教授负责撰写，其余章节由唐培和教授负责撰写。唐新来教授审阅了部分章节。

本书的出版得到了广西教育厅特色专业及课程一体化项目建设经费（GXTSZY217）、广西高校名师专项经费、广西高等教育教学改革工程重点项目（2014JGZ133）经费的资助。电子工业出版社的有关同志为本书的出版做了大量的工作。在此一并表示感谢。

本书综合了作者多年的教学经验，也吸取了同行很多学者的成果。参考文献中列出了部分引用的文献，恐有遗漏（特别是网上的文献），敬请谅解！

由于作者水平有限，时间仓促，难免错漏。不当之处，请读者批评指正。欢迎全国同行交流经验与体会，联系方式：tangpeihe@163.com。

本书为教师提供相关教学资源，有需要者，请登录华信教育资源网 <http://www.hxedu.com.cn>，注册之后进行下载。

作 者

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为以及歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，本社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396; (010) 88258888

传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市海淀区万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036

目 录

第 1 章 数据结构与算法概述	1
1.1 引言	1
1.1.1 《数据结构与算法》课程到底研究什么	1
1.1.2 《数据结构与算法》课程介绍	1
1.1.3 为什么要学习《数据结构与算法》课程	2
1.2 基本概念和术语	3
1.2.1 数据	3
1.2.2 数据项	3
1.2.3 数据元素	4
1.2.4 数据对象	4
1.2.5 数据结构	4
1.2.6 数据结构的形式化描述	6
1.2.7 数据的逻辑结构、存储结构与物理结构	7
1.2.8 数据结构实例	7
1.3 算法	9
1.3.1 什么是算法	9
1.3.2 算法的性质	11
1.3.3 算法和程序	12
1.4 算法的表示（描述）	14
1.4.1 自然语言	14
1.4.2 计算机语言	15
1.4.3 图形化工具	16
1.4.4 伪代码	17
1.5 算法设计的准则	19
1.5.1 正确性	19
1.5.2 可读性	20
1.5.3 健壮性	20
1.5.4 效率	21
1.6 算法效率分析	24
1.6.1 事后统计的方法	25
1.6.2 事前分析估算的方法	25
1.6.3 算法的空间复杂度	28
1.7 抽象数据类型	30

阅读材料	32
习题 1	35
第 2 章 顺序存储结构的线性表	39
2.1 基本概念和操作	39
2.1.1 什么是线性表	39
2.1.2 线性表的逻辑结构	40
2.1.3 线性表的基本操作	40
2.1.4 线性表的顺序存储结构	41
2.1.5 线性表的插入与删除操作（算法）	42
2.1.6 顺序表应用举例	45
2.2 堆栈	47
2.2.1 堆栈的定义	47
2.2.2 栈的顺序存储结构与操作（算法）	48
2.2.3 堆栈在算法设计中的应用	51
2.3 队列	63
2.3.1 队列的定义	64
2.3.2 队列的顺序存储结构与操作（算法）	64
2.3.3 循环队列	66
2.3.4 队列应用举例	68
2.4 集合及其运算	71
2.4.1 集合的定义	71
2.4.2 集合运算	72
2.4.3 集合的顺序存储结构和操作实现	72
小结	75
习题 2	75
第 3 章 链式存储结构的线性表	79
3.1 什么是链式存储结构的线性表	79
3.2 线性链表的操作（算法）	82
3.2.1 线性链表的构造	83
3.2.2 求表长	86
3.2.3 查找操作	86
3.2.4 线性链表的插入	87
3.2.5 线性链表的删除	89
3.3 栈的链式存储结构	90
3.4 队列的链式存储结构	92
3.5 循环链表	93
3.6 双向链表	94

3.7 静态链表	95
3.8 链表应用	97
3.9 一元多项式的存储和相加	100
3.10 集合的链式存储结构与操作	103
3.11 顺序表和链表的比较	106
习题 3	107
第 4 章 串	110
4.1 基本概念	111
4.2 串的存储结构	112
4.2.1 顺序存储结构	112
4.2.2 链式存储结构	113
4.3 串的基本操作	113
4.3.1 串复制	114
4.3.2 求字符串的长度	115
4.3.3 字符串连接	115
4.3.4 取子串	115
4.3.5 判断两个字符串是否相等	116
4.3.6 插入字符串	117
4.3.7 删除子串	117
4.3.8 字符串清空	118
4.3.9 判断字符串是否为空	118
4.3.10 字符串比较	118
4.4 串的模式匹配算法	119
4.4.1 模式匹配及其 BF 算法	119
4.4.2 模式匹配的一种改进算法——KMP 算法	120
4.4.3 其他模式匹配算法	124
4.5 串操作应用举例	132
习题 4	133
第 5 章 数组、特殊矩阵与广义表	136
5.1 数组的定义和运算	136
5.1.1 数组的基本概念	136
5.1.2 数组的特点	137
5.2 数组的顺序存储结构	137
5.2.1 一维数组	137
5.2.2 二维数组	138
5.2.3 三维数组	138
5.2.4 n 维数组	139

5.3	特殊矩阵的压缩存储与处理	141
5.3.1	基本原则	141
5.3.2	对称矩阵的压缩存储	141
5.3.3	三角矩阵的压缩存储	142
5.3.4	对角矩阵(带状矩阵)的压缩存储	143
5.4	稀疏矩阵	144
5.4.1	稀疏矩阵的三元组表存储	145
5.4.2	稀疏矩阵的转置	145
5.4.3	稀疏矩阵的快速转置算法	147
5.4.4	稀疏矩阵的乘积	148
5.4.5	稀疏矩阵的十字链表存储	151
5.5	广义表	155
5.5.1	广义表的定义	155
5.5.2	广义表的特性	156
5.5.3	广义表的基本操作	156
5.5.4	广义表的存储	157
5.5.5	广义表的基本操作	159
	习题 5	162
第 6 章	树与二叉树	166
6.1	基本概念	166
6.1.1	什么是树	166
6.1.2	基本概念和术语	167
6.1.3	树的表示方法	167
6.1.4	树的基本操作	168
6.1.5	树的存储结构	168
6.2	二叉树	173
6.2.1	二叉树的定义	173
6.2.2	二叉树的基本形态	173
6.2.3	两种特殊的二叉树	174
6.2.4	二叉树具有的重要性质	175
6.2.5	二叉树的存储结构	176
6.2.6	二叉树的基本操作	179
6.3	二叉树的遍历	179
6.3.1	二叉树的遍历	179
6.3.2	先序遍历算法	180
6.3.3	中序遍历	181
6.3.4	后序遍历	182

6.3.5	按层次顺序遍历二叉树	183
6.3.6	遍历算法的应用	184
6.4	二叉树其他运算	185
6.4.1	建造二叉树算法	185
6.4.2	求二叉树高度(深度)	186
6.4.3	求二叉树宽度	186
6.4.4	二叉树查找	187
6.4.5	输出一棵二叉树	188
6.4.6	清除一棵二叉树	188
6.5	线索二叉树	189
6.5.1	建立线索树	189
6.5.2	检索结点	191
6.5.3	在线索二叉树上插入一个结点	192
6.6	树与森林	193
6.6.1	树与二叉树的转换	193
6.6.2	森林与二叉树的转换	194
6.6.3	树和森林的遍历	194
6.7	树的应用	197
6.7.1	判定树	197
6.7.2	集合的表示	198
6.7.3	关系等价求等价类问题	199
6.8	二叉排序树	200
6.8.1	二叉排序树的定义	200
6.8.2	二叉排序树的构造	201
6.8.3	二叉排序树的查找	203
6.8.4	二叉排序树的更新	205
6.8.5	二叉排序树的删除	205
6.9	哈夫曼树及其应用	207
6.9.1	基本概念	207
6.9.2	哈夫曼树的构造	208
6.9.3	哈夫曼编码	210
6.9.4	最佳判定过程	211
6.10	平衡二叉树	212
6.10.1	平衡二叉树的引入	212
6.10.2	平衡二叉树的定义	212
6.10.3	最小不平衡二叉树	213
6.10.4	不平衡二叉树的调整方法	214

6.10.5 建立平衡二叉树举例	218
6.10.6 平衡二叉树与二叉排序树比较	219
习题 6	220
第 7 章 图及其应用	223
7.1 基本概念和术语	225
7.1.1 图的定义	225
7.1.2 完全图、稠密图、稀疏图	226
7.1.3 顶点的度	227
7.1.4 子图	227
7.1.5 路径和回路	227
7.1.6 连通和连通分量	227
7.1.7 强连通图和强连通分量	227
7.1.8 权和网	228
7.2 图的存储结构	229
7.2.1 邻接矩阵	229
7.2.2 邻接表	231
7.2.3 十字链表	234
7.2.4 邻接多重表	236
7.2.5 图的边集数组	237
7.3 图的遍历	237
7.3.1 深度优先搜索	237
7.3.2 广度优先搜索	239
7.3.3 搜索算法在搜索引擎中的应用	242
7.4 图的连通性	243
7.4.1 无向图的连通性	243
7.4.2 有向图的连通性	244
7.4.3 生成树和生成森林	245
7.4.4 关节点和重连通分量	245
7.5 最小生成树	248
7.5.1 什么是生成树	248
7.5.2 构造最小生成树的普里姆 (Prim) 算法	248
7.5.3 构造最小生成树的 Kruskal 算法	252
7.6 最短路径	255
7.6.1 求某一个源点到其他顶点的最短路径	256
7.6.2 Bellman-Ford 算法	259
7.6.3 每对顶点之间的最短路径	262
7.7 有向无环图及其应用	267

7.7.1 有向无环图的概念	267
7.7.2 AOV 网与拓扑排序	268
7.7.3 AOE 网与关键路径	272
习题 7	277
第 8 章 查找与搜索引擎	281
8.1 基本概念与术语	281
8.2 顺序查找	283
8.3 二分查找(折半查找)	285
8.3.1 二分查找的定义	285
8.3.2 二分查找算法	286
8.3.3 二分查找判定树	287
8.3.4 二分查找性能分析	287
8.4 有序表的插值查找和斐波那契查找	288
8.4.1 插值查找	288
8.4.2 斐波那契查找	288
8.5 分块查找	289
8.6 B-树和 B+树上的查找	291
8.6.1 B-树及其查找	291
8.6.2 B+树	293
8.7 哈希表与散列查找	294
8.7.1 散列的概念	295
8.7.2 哈希函数	295
8.7.3 散列存储的冲突	298
8.7.4 装填因子	298
8.7.5 冲突解决方法	298
8.7.6 散列存储的平均查找长度	302
8.7.7 散列存储的优缺点	303
8.8 搜索引擎及其相关技术	303
8.8.1 网页的自动下载与存储	304
8.8.2 网页索引技术	306
8.8.3 网页排名技术	310
习题 8	323
第 9 章 排序	326
9.1 基本概念	327
9.2 插入排序	327
9.2.1 直接插入排序	327
9.2.2 折半插入排序	329

9.2.3	表插入排序	330
9.2.4	希尔排序	332
9.3	交换排序	336
9.3.1	冒泡排序	336
9.3.2	快速排序	337
9.4	选择排序	343
9.4.1	简单选择排序	343
9.4.2	树形选择排序	346
9.4.3	堆排序	350
9.5	归并排序	356
9.6	分配排序	359
9.6.1	箱排序	359
9.6.2	基数排序	360
9.6.3	多关键字排序	363
9.7	外排序	364
9.7.1	外部排序的方法	364
9.7.2	多路归并的实现	365
9.8	排序方法之比较	368
习题 9		369

第 1 章

数据结构与算法概述

1.1 引 言

1.1.1 《数据结构与算法》课程到底研究什么

要想回答这个问题,我们必须从宏观上了解这样一个情况:利用计算机解决客观世界的问题时,人们需要做什么?怎么做?为了便于地理解,先看图 1-1。

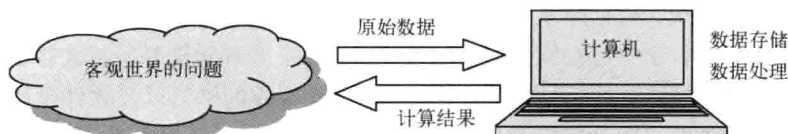


图 1-1 基于计算的问题求解模型

从图 1-1 可以看出,利用计算机解决客观世界中的问题至少需要研究解决 4 方面的问题:① 输入数据;② 数据存储;③ 数据处理;④ 输出结果。

首先,输入数据主要解决原始数据(原始数据相当于工厂企业的原材料)的输入问题。这个问题看似简单,其实不然,特别是面对大规模、多类型数据输入以及自动感知(视觉、听觉、嗅觉、触觉等)时,难度非常大。

其次,数据进入计算机以后,需要解决的问题是存储,即存什么地方以及怎么存放。就像企业的原材料进了工厂后放哪里呢?怎么存放呢?如何保管呢?……

数据存储问题解决后,接下来要做的事情就是数据处理(计算),这相当于企业对原材料的加工处理。只有加工处理后,才能得到外面所需要的计算结果(相当于企业生产的半成品或成品)。

最后需要做的是输出计算结果,这是我们最终需要的答案。输出结果也不见得是一件很容易的事情,关键看具体需求了。例如,人们有时候需要给出直观的图形作为问题的解,有限元分析与计算就是比较典型的代表。

《数据结构与算法》课程关心的是中间两个环节的内容,即数据存储和数据处理。至于输入数据和输出结果方面的知识则点到为止,相关技术和方法不做详细介绍(其他课程会介绍)。

1.1.2 《数据结构与算法》课程介绍

1968 年,美国的唐纳德·克努特¹(Donald Ervin Knuth)教授在其所写的《计算机程序设计艺术》

¹计算机界绝对的牛人,译名很多,以后干脆称之 Knuth。由于 Knuth 为《数据结构与算法》的鼻祖,因此本章后面以阅读材料的形式专门介绍他。