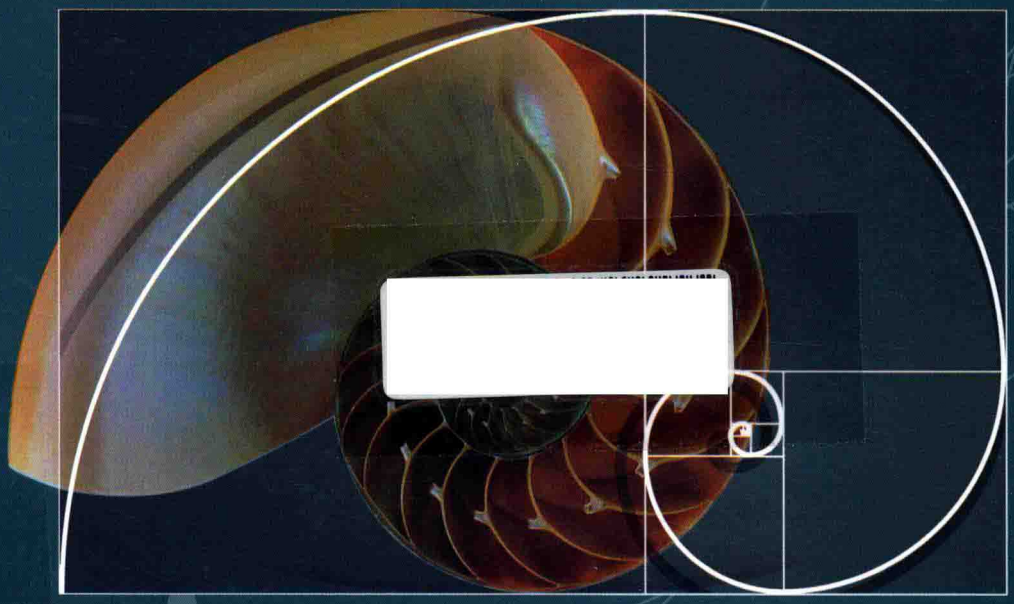
数据结构与算法分析新视角

周幸妮　任智源　马彦卓　樊　凯　编著



中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

数据结构与算法分析新视角

周幸妮　任智源　马彦卓　樊　凯　编著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

数据结构是高等学校计算机及其相关专业的核心课程，是计算机程序设计的基础。本书按照“像外行一样思考，像专家一样实践”的解决问题的思维方法，列举大量实际或工程案例，从具体问题中引出抽象概念，运用类比、图形化描述等各种方式，对经典数据结构内容做深入浅出的介绍。在介绍数据结构和算法的基本概念和算法分析方法的基础之上，从软件开发的角度，通过应用背景或知识背景介绍、数据分析、函数设计、算法设计、测试调试等环节，分别对顺序表、链表、栈、队列、串、数组、树、图等基本类型的数据结构进行了分析和讨论；介绍数据的典型操作方法，如数据排序方法和查找方法；介绍常见的如递归法、分治法、动态规划、贪心法等经典算法。

本书可作为计算机及相关专业大学本科教材，也可作为应用型专业以及成人教育、工程技术人员的培训教材或自学教材。

图书在版编目 (CIP) 数据

数据结构与算法分析新视角 / 周幸妮等编著. —北京：电子工业出版社，2016.4
ISBN 978-7-121-28084-9

I. ①数… II. ①周… 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（2016）第 011970 号

策划编辑：窦　昊
责任编辑：窦　昊
印　　刷：三河市鑫金马印装有限公司
装　　订：三河市鑫金马印装有限公司
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：787×1 092　1/16　印张：33.5　字数：858 千字
版　　次：2016 年 4 月第 1 版
印　　次：2016 年 4 月第 1 次印刷
定　　价：69.00 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

从新视角来看待旧问题，则需要创造性的思维——爱因斯坦

数据结构的教与学经历

六七年前上数据结构课时，驾轻就熟地依然按照一贯的讲法上课。上了几次课后，收到班上一位学生的 E-mail，信中说："我自己是特别热爱写程序的。一方面我很熟悉电脑，软硬件都有涉猎，所以学起来就不缺基础的相关知识（像内存、寄存器、电子电路等等这些都很熟悉）；另一方面我好像很能适应，也很喜欢这种思维方式……但好像班上不少的同学对数据结构的学习理解和接受起来还是比较困难的……"

教授数据结构的课程也有十余年了，一直以来，学生们总是认为数据结构不是一门容易学的课程，"在众多的专业课中，数据结构被很多学生认为是一门很难学习的课程。"[1]虽然我在学校读书时没有学过数据结构的课程，只是因为后来要教书，才自学的数据结构。在自学的过程中，也并没有觉着内容怎么难啊，这是怎么回事呢？

仔细回想一下自己学习与工作的经历过程，或许就是来信的这位学生说的，是因为在学习数据结构书本知识之前，已经有了较强的编程的技能、一些数据结构实际应用的先验知识吧。比如，在研究所工作时首次参加的软件开发项目中，就有多进程、链表、队列、散列等的实际应用，虽然在学校并没有学过这些概念，而是先接触到实际项目中要处理的问题，再看到其他程序员的具体算法处理和程序实现方法，从实际的问题切入，就比较容易理解相应的数据组织形式和对应的算法，等到后来再接触到书本的理论知识，就有一种"一通百通，豁然开朗"的感觉。还有一个原因是在软件开发过程中逐渐熟悉并掌握了程序调试方法，对例程通过跟踪的方法，很容易弄清执行的路径和结果，对算法的设计和实现的理解也起到了事半功倍的效果。

回头来看学生们学习的教科书，概念的介绍是传统意义上的叙述方式，抽象度很高，但从实际到抽象、从抽象到实际的过程介绍不足，即感性认识不足，抽象就难以理解接受。"现在有一个不好的倾向，就是教材或课堂过于重视抽象化的知识，忽视应用背景。数据结构的教材是这一倾向的代表。这对入门阶段的学生来讲是不适宜的，因为学生难以走进所涉及的抽象世界，最终表现为不知道在讲什么。"[2]

再看我的学生，既没有实际软件开发的经验也没有熟练的编程调试基础，对数据结构结构没有感性认识，先接触的是那些抽象的概念，感到理解和接受困难也就可以理解了。邹恒明在《数据结构：炫动的 0、1 之弦》一书中指出，对于很多人来说，数据结构的概念并不难，真正的难点是：

（1）如何实现从数据结构概念到程序实现的跨越（即如何实现一个数据结构）；

（2）如何实现从实际应用到数据结构抽象的跨越（即如何利用数据结构解决实际问题）。

对于我来说，仅仅在学校学了一点点程序设计语言（记得所有上机时间加起来不超过 20 小时），没有任何数据结构的知识，刚出校门就参与了历时三年多后来获得国家科技进步三等奖的大型软件开发工作，以及后续多个电信用户单位的实际软件安装、应用调试和维护工作，亲历了实现上述两个“跨越”的最实际生动的案例。虽然项目的开发过程非常艰苦，有在用户单位调试现场连续大半年的天天加班到深夜、第二天依然要正点到机房的超负荷工作，有通宵的跟踪调试，有 24 小时在线系统内存泄漏的巨大压力，有上线后双机备份系统同时崩溃争分夺秒找 bug 的惊心动魄，等等。应该说自己是很幸运的，虽然在学校仅仅学习了一点点编程的概念，但在工作中根据需要自学和向同事学习了很多新知识、经验和技巧，这样的实践磨练，对后来的程序设计类课程的理解和教授，是非常有益的。

学习数据结构困难的症结所在

回想与总结起来，之所以要有上述两个鸿沟要“跨越”，也是由于学校的传统教科书教法和实际的应用要求脱节造成的。

弗里德里希·威廉·尼采曾写道：“人们无法理解他没有经历过的事情。”[3]换句话来说，我们只接受与过去早已理解的事物相关的信息。这是一种比较学习的过程，在这个过程中，大脑要寻找每条信息之间的联系，借助以往经验来理解新事物[4]。

“欧拉认为，如果不能把解决数学问题背后的思维过程教给学生的话，数学教学就是没有意义的。现代计算机实质上的发明者莱布尼兹也说过：在我看来，没有什么能比探索发明的源头还要重要，它远比发明本身更重要。从小到大，我们看过的数学书几乎无一不是欧几里德式的：从定义到定理，再到推论。这样的书完全而彻底地扭曲了数学发现的真实过程。目前几乎所有的算法书的讲解方式也都是欧几里德式的，每一个推导步骤都是精准制导直接面向数据结构目标的，实际上，这完全把人类大脑创造发明的步骤给反过来了。对读者来说，这就等于直接告诉你答案和做法了，然后让你去验证这个答案和做法是可行或成立的，而关于答案和做法到底是怎么来的，从问题到答案之间经历了怎样的思维过程，却鲜有书能够很好地阐释。对于这类知识讲述（欧几里德方式）方式的批判，西方（尤其是在数学领域）早就有了。”[5]

传统数据结构教科书的一般模式都是给出问题，然后直接给出算法，而实际上要用计算机解决问题，必须要考虑的处理步骤有：如何分析问题中的已知信息，如何提炼数据及数据间的联系（数据的逻辑结构），如何选用合适的存储方式（数据的存储结构）将逻辑结构存到计算机中，然后在存储结构之上按照自顶向下逐步细化的方法给出算法，这才是真正解决实际问题的思维方法和步骤，也是软件开发中实际采用的方法。传统教科书的问题在于没有一个思维过程的引导与分析，致使概念论述、实现细节有余，设计实现过程描述不足，让学生看到的只是一个个问题具体的详解，而把握不住算法设计的总方法和原则。

本书尝试从学以致用的角度，注意给出问题或算法的知识背景或应用背景，增加一些在实际软件开发中的算法应用背景或实例；强调算法的分析方法、设计思路、给出重要算法的测试及调试分析，弥补上述传统教科书中的不足。教学生以“软件开发的方法”处理问题，使学生容易理解并掌握它，在实际的软件开发过程中能灵活地选择适当的逻辑结构、存储结构及相应的算法，设计性能优、效率高、可读性强、易维护的程序，达到数据结构课程最终的目的。

程序设计与数据结构的关系

我们在学习数据结构知识之前要有程序设计的基础，那么我们先来看看与编程相关的问题。

什么是编程？编程不仅仅是对语法的掌握，还涉及下面的诸多方面。

（1）程序的解题思路——算法是基本运算及规定的运算顺序所构成的完整的解题步骤，是程序的灵魂，算法的优劣直接影响程序的效率。本书的算法描述方法见稍后的说明。

（2）程序的运行速度——程序运行的速度受很多因素的影响。用户对程序的运行速度往往是有要求的，如实时响应系统。

（3）程序的运行空间——代码运行需要相应的内存空间及相关运行环境。在有些应用场合，对程序占用空间是有限制的，如嵌入式系统。

（4）代码规范——代码要按照一定的规范格式书写，以保证代码的一致性，便于交流和维护。

（5）程序的结构—— 一个功能复杂的程序由多个功能相对独立的模块组成。模块内高内聚，模块间低耦合，是判断程序结构是否合理的标准。

（6）模块接口——模块间的信息交流通过接口完成，模块间信息传递参数的设置应该合理有效。

（7）程序的测试与调试——要有精心设计的测试样例来测试程序是否正确。调试是高效率完成软件产品的有效方法。一个程序高手，也是调试专家，调试的经验方法多数都是实践中得到的。

我们在学习数据结构知识之前要有程序设计的基础，那么数据结构与程序设计间的关系是怎样的呢？应该说数据结构是编写规模庞大、逻辑复杂的更高级程序所需的基础。表 0.1 给出了程序设计与高级编程的特点。

表 0.1　程序设计与高级编程

	涉及课程	主要内容	课程目标
结构化程序设计	程序设计基础类课程	语言语法形式、语句使用规则 模块设计思想	编写简单程序 解决简单问题
高级编程技术	数据结构	数据的抽象思维方法	编写规模庞大、逻辑复杂的程序
		算法的规范声明、算法的性能分析、算法的性能评价	
	软件工程	部件（模块）设计思想、软件工程思想	
	……		

程序设计的首要问题是模块划分及相关问题，另一个重要方面，是把要解决问题的信息转换成计算机能认识并接收的数据，这一转换过程就是数据的抽象过程。要处理规模庞大的复杂问题，必须掌握数据的抽象思维方法，同时还要熟练掌握算法的规范声明、算法的性能分析、算法的性能评价等诸多技能。

数据结构与其他课程的关系

作为一门重要的专业核心必修课程，“数据结构与算法”课程既是对以往课程的深入和扩展，也为深入学习其他专业课程打下基础。课程中排序问题算法以及基本的树、图等数据结构，是计算机科学的基本功。B+树、散列（Hash）等高级数据结构，也是数据库、操作系统、编译原理、计算机网络等重要专业课程的基础。本课程在计算机学科中与其他课程的关系如图 0.1 所示。

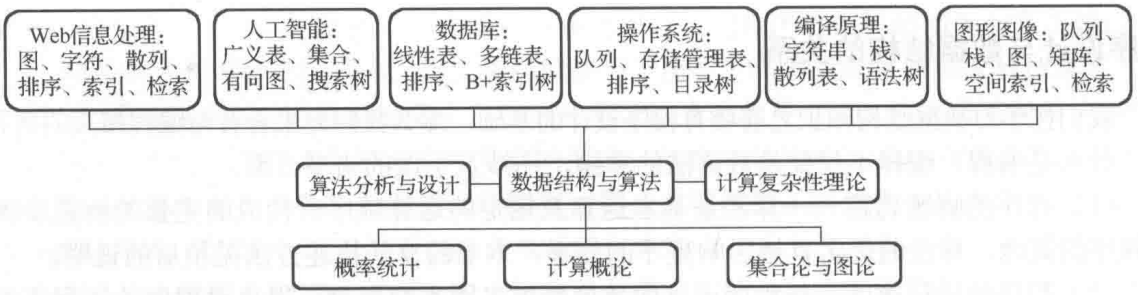


图 0.1 “数据结构与算法”课程在计算机学科中的重要地位

数据结构的重要性

商用程序员肖舸在他的博客中写道：“这么多年，我做过游戏、通信、工业控制、教育、VoIP、服务器集群等各个方向的项目，不可谓不宽。

但是我知道的是，其实我都是在用一种方法写程序。那就是从最底层的数据结构和算法开始做起，用最基本的C、C++语言开发。用来用去，还是那么几个数据结构，队列、堆栈，等等。

这好比武侠小说里面的内功，内力修好了，学招式，非常容易。但如果没有内力，练再好的招式，见到高手就软了。一力破十慧，就是这个道理。在绝对的实力面前，任何花招都是没有用的。”[6]

对清华大学计算机系历届毕业生和部分研究生追踪调查显示，几乎所有的学生都认为“数据结构”是他们在学校里学过的最有用的课程之一；数据结构是国内外许多软件开发机构要求考核的基本课程之一；IT 业公司面试或笔试考核的绝大部分内容是数据结构或算法；数据结构也是计算机科学与技术、软件工程等专业研究生考试必考科目。

数据结构会过时吗

电子计算机自 20 世纪 40 年代诞生以来硬件上不断更新换代，软件也是同步发展的，作为软件的一个分支——程序设计语言也从机器语言发展到了第四代，但无论软件如何发展，无论开发工具如何进步，只要我们的计算机还是基于冯·诺依曼体系，数据结构和算法仍然是程序的核心，永远不会被淘汰。

学习数据结构和算法并不仅仅要求我们学会如何使用和实现某种数据结构，更重要的是学会分析问题、解决问题的思想和方法。

本书关于算法与程序实现增加的内容

在基础数据结构——线性表与运算受限的线性表栈与队列章节中，特别增加了下列内容。

1. 增加测试样例的设计

从程序健壮性的角度出发，测试样例的设计是开始程序设计就应该考虑的一个重要内容。一般的程序设计与数据结构教科书很少考虑测试样例问题，本书在最初基本的算法设计中，给出了基础程序的测试样例设计，也是让学习者以专业的方法学习程序设计，养成良好的设计习惯。

2. 增加函数结构的设计

对初学者而言，在学习数据结构时，程序设计语言基础并不扎实，特别是函数结构的设

计不熟练。根据给定的功能、输入/输出信息，在调用与被调用的函数关系中，信息是怎样传递的，往往存在多种可能的选择，该怎样确定合适的形参类型、函数类型，初学者经常会无所适从，从而造成编程困难。根据这种情况，本书特别给出了典型数据结构运算的多种方案实现，在同一问题不同函数结构设计的比较中，让读者发现各种信息传递的方式和特点，巩固和熟练编程知识和技巧，达到理论与实践的统一。

函数结构样例分别见表 0.2 和表 0.3。

表 0.2　函数结构样例 1

功能描述	输入	输出
顺序表中查找给定的值 LocateElem_Sq	顺序表地址 SequenList *	正常：下标值 int
	结点值 ElemType	异常：-1
函数名	形参	函数类型

表 0.3　函数结构样例 2

功能描述	输入	输出	
顺序表中取给定下标的元素值 GetElem_Sq	顺序表地址 SequenList *	返回	正常：1
	下标 int		异常：0
	（结点值 ElemType）	结点值 ElemType	
函数名	形参	函数类型	

说明：当函数中的输入和输出项中的内容一样时，表示函数的返回信息是通过形参的址传递方式，而非 return 方式。如表 0.3 中函数结构样例 2 中，输入/输出都有“结点值 ElemType”项，表示形参采用“地址传递”方式。

3．增加重要程序的调试

程序设计的过程也是一个测试与调试的过程，程序的编与调是密不可分的。对于初学者而言，若调试不熟练，容易丧失继续学习的信心。根据多年的教学经验，若有程序调试演示，则相关程序比较容易掌握，特别是在学习链表等内容时，光是解释数据结构、结点联系，学生总觉得抽象难懂，若通过演示调试，观察各结点间的联系如何动态建立、消除等，则生动直观，一目了然。由于调试过程步骤较多，学生看过了明白了，但要再回想并模拟跟踪过程，则是不容易的事情，所以本书把一些重点例子的程序跟踪过程记录下来，以便学生学习。本书所有的源码均以 C 语言编制，并在 VS IDE 环境下通过调试和测试。

4．算法描述方法

算法描述按照下面的顺序给出（主要用于编程最基础的线性表部分），内容详见 1.5.3 节。

1）测试用例设计。

2）数据结构描述。

3）函数结构设计。

4）算法伪代码描述。

5）程序实现。

6）算法效率分析：

注：#define TRUE 1

#define FALSE 0

数据结构课程简介

用计算机解决实际问题，首先要做的事情就是要把涉及问题的相关信息存储到计算机中，也就是需要把问题的信息表示为计算机可接收的数据形式，然后根据问题处理功能的要求，对存储到计算机中的数据进行处理。归结为一句话，可以这样说，用计算机解题首先要用合理的结构表示数据，然后才能根据相应的算法处理数据，而数据表示和数据处理正是数据结构学科要研究的内容。

“数据结构”主要介绍如何合理地组织数据、有效地存储和处理数据，正确地设计算法以及对算法的分析和评价。

通过本课程的学习，学生能透彻地理解数据结构的逻辑结构和物理结构的基本概念以及有关算法，学会分析数据对象特征，掌握数据组织方法和计算机的表示方法，以便在实际的软件开发过程中灵活地选择适当的逻辑结构、存储结构及相应的算法，设计性能优、效率高、可读性强、易维护的程序，解决实际问题。

数据结构是实践性很强的一门课程，本课程的特色就是注重实践活动。通过大量上机实验，能加深对课程理论知识的理解，增强学习的兴趣，提高计算机计算法思维逻辑能力，为以后的学习和工作打下坚实的基础。

本课程的基本要求如下：

- 了解数据结构及其分类、数据结构与算法的密切关系。
- 熟悉各种基本数据结构及其操作，学会根据实际问题要求来选择数据结构。
- 掌握设计算法的步骤和算法分析方法。
- 掌握数据结构在排序和查找等常用算法中的应用。
- 初步掌握索引技术。
- 了解经典算法。

本书的组织结构

本书共分 9 章。第 1 章介绍数据结构和算法的基本概念和算法分析方法。第 2 章至第 6 章从软件开发的角度，通过应用背景或知识背景介绍、数据分析、函数设计、算法设计、调试等环节，分别对顺序表、链表、栈、队列、串、数组、树、图等基本类型的数据结构进行分析和讨论。第 7 章排序技术介绍常见的数据排序方法，第 8 章索引查找技术介绍数据的查找方法，第 7、8 章的内容是对数据的典型操作方法的介绍。第 9 章经典算法，介绍一些常见的如递归法、分治法、动态规划、贪心法等经典算法。

前言、第 1 章至第 6 章由周幸妮老师完成；第 7、9 章由任智源老师完成；第 8 章由马彦卓老师完成；樊凯老师提供了通信等方面的部分专业应用案例；周幸妮对全书进行了统稿。

书中设有“思考与讨论”、“知识 ABC”等栏目。在“思考与讨论”栏目中提出问题，引起读者思考，并对提出的问题进行探讨，帮助读者加深对相关概念等的理解，以期活跃思路、扩展思维。“知识 ABC”栏目主要介绍相应概念的知识、知识背景、算法的发明背景或故事等，以扩大读者的知识面和了解数据结构的应用背景等。有些知识可能专业性比较强，对此可以“不求甚解”，大致了解即可，感兴趣了可以再去查相关资料。

致谢

本书的写作动力，依然是源自于我的学生，起因是09级教改班学生的提议，认为我的授课方式、思路都不错，写出来会别具一格，让大家学习这个课程时能更好理解一些，这部分内容在我的《C语言程序设计新视角》一书中有所提及。

我的动力亦源于父亲的鼓励、家人的支持。感谢铁满霞教授、陈慧婵教授的指点；感谢学生屈宇澄自始至终的支持和帮助；感谢孙蒙、丁煜、孙舒、孙亚萍、张柯、杨恒杰、吴伟基、黄超、张平等学生的热心帮助；感谢通信工程学院13级教改班、14级卓越班，空间科学与技术学院13级实验班学生们对本书初稿讲义提出的各种有益的意见和建议。

所有和同学们、同事们的讨论都让人受益匪浅，这些讨论或开拓了视野，或引起了深入思考，所有的一切都是一种成长与完善的历程。

感谢曾经给予我很多帮助、一起辛苦工作的同事们；感谢我有趣的人生经历；感谢我可爱的学生们；感谢所有关心和帮助我的人们。

感谢 Bandari 的 April，每天伊始，美妙的乐曲让人在纯净、安定、温暖的情绪中开始一天的写作。

感恩一切。

周幸妮

xnzhou@xidian.edu.cm

2015年春于长安

[1] 邹恒明.《数据结构：炫动的0、1之弦》

[2] 叶常春.“程序设计语言”课程教学方法探讨《计算机教育》2007年21期

[3] 弗里德里希·威廉·尼采（Friedrich Wilhelm Nietzsche，1844—1900），德国著名哲学家

[4] 里斯.特劳特.《新定位》，中国财政经济出版社

[5] 刘未鹏.《知其所以然，以算法学习为例》（网上专栏）

[6] http://student.csdn.net/space.php?uid=39028&do=thread&id=198&page=1

目　录

第1章 绪 论

【主要内容】

- 数据结构的概念
- 算法设计的基本要求
- 从时间和空间角度分析算法效率的方法

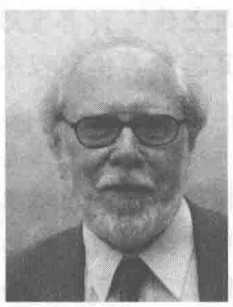
【学习目标】

- 了解数据结构课程的意义
- 掌握数据结构的概念
- 掌握算法设计与程序设计的步骤
- 掌握算法效率的分析评价方法

1.1 从编程说起

学过程序设计的人都知道，用人脑驱动计算机的方式是编程。瑞士著名的计算机科学家，大名鼎鼎的 Pascal 之父尼古拉斯·沃斯（Niklaus Wirth）教授就编程的概念，提出了一个著名的公式，并由此获得了计算机科学界的最高荣誉“图灵奖”，公式仅仅只有一行文字：

Algorithm（算法）+ Data Structures（数据结构）= Programs（程序）



Niklaus Wirth 1934.2.15—

Nicklaus Wirth 在 1976 年出版了一本书，书名即为《算法+数据结构 = 程序设计》，它阐述了数据结构在程序设计中的作用。编程前首先需要解决两个问题：算法设计和数据结构设计。算法是处理问题的策略，而数据结构是描述问题信息的数据模型，程序则是计算机按照处理问题的策略处理问题信息的一组指令集。

我们设计程序的目的是让计算机帮助人们自动地完成所要处理的复杂任务，计算机科学与技术的根本问题——什么能够被自动化以及有效地自动化？具体为由实际的问题到最终的计算机求解，要经过怎样的过程？数据结构与算法在其中的作用是什么？

要通过计算机来解题，一般性步骤如图 1.1 所示。图中矩形框中是问题处理的阶段或结果，椭圆框为处理过程，其中：

- 具体问题：问题中要有已知条件及实现要求的描述。
- 问题模型：通过对问题进行定性定量分析，从中提取出要完成的功能、要处理的信息，并找出这些信息之间的关系，提炼出问题的两个要素——信息与功能，建立问题模型。建立问题模型是为了将实际问题转化为计算机能“理解”并能“接收”的形式。
- 数据结构：分析问题模型中信息里包含的数据是什么、数据间的联系是什么、以什么形式存储在计算机中，分析后形成数据结构。

- 算法：将问题中输入输出是什么，功能是什么，实现的步骤有哪些，算法的速度怎样，占的空间是多少等进行分析与设计之后形成处理方案。
- 程序：程序设计将算法“翻译”成相应命令语句及处理形成代码。
- 问题得解：可以通过各种专业测试方法对程序进行测试，测试合格最终得到问题的解。

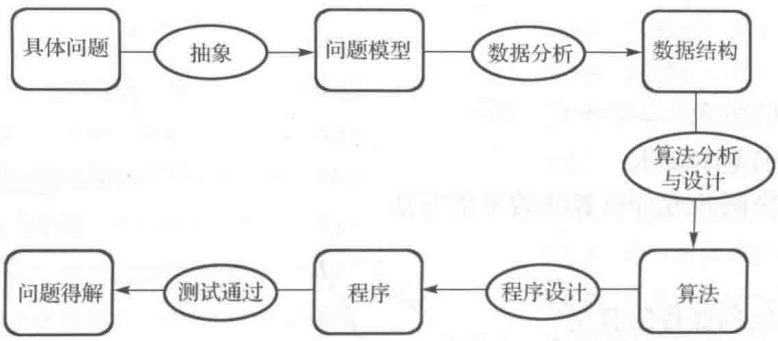


图 1.1 计算机解决问题的过程

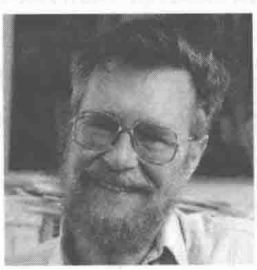
【思考与讨论】从程序设计的角度看，“把数据存入计算机并处理”是如何实现的？

讨论：

周以真 Jeannette M. Wing

美国卡内基·梅隆大学周以真教授在“计算思维”概念的阐述中说：“计算思维，是用计算的基础概念去求解问题、设计系统、理解人类行为。计算思维是建立在计算过程的能力和限制之上的”。计算思维最根本的内容，即其本质是抽象和自动化。计算思维具体到程序设计中，可以用以下的“程序的思维”来描述。

程序设计中信息的抽象是用标识符、常量、变量、数组和结构体等描述和记录信息及信息间的关系；自动化是用语句及运算符按预设的目的操纵处理信息；语句的组织结构按照功能的独立性划分即是函数。一个大问题分解为多个子问题，相互独立又相互联系的关系是通过函数来完成的；算法则描述了操纵处理的方法和步骤，适合人的思维方式的算法描述方法是按照自顶向下逐步求精的方法来进行的。这些组合在一起构成了程序设计语言和程序设计方法。



Edsger W. Dijkstra
1930.5.11—2002.8.6

【知识 ABC】“自顶向下逐步求精”程序设计方法

结构化程序设计（Structured Programming）是进行以模块功能和处理过程设计为主的详细设计的基本原则。其概念最早由迪杰斯特拉（E. W. Dijkstra）在 1965 年提出的，是软件发展的一个重要的里程碑。主要观点是采用自顶向下、逐步求精的程序设计方法；使用三种基本控制结构构造程序，任何程序都可由顺序、选择、循环三种基本控制结构构造。以模块化设计为中心，将待开发的软件系统划分为若干个相互独立的模块，这样使完成每一个模块的工作变得单纯而明确，为设计一些较大的软件打下了良好的基础。

自顶向下，指程序设计时，应先考虑总体，后考虑细节；先考虑全局目标，后考虑局部目标。不要一开始就过多追求众多的细节，先从最上层总目标开始设计，逐步使问题具体化。逐步细化，是针对复杂问题，设计一些子目标作为过渡，逐步细化。

要开发复杂的软件，需要有科学的构建和维护软件的方法，即把做软件当做一项工程项目来处理，这涉及软件工程学科的内容。从软件工程的观点来看程序的开发过程，它是由多个阶段构成的，依次是需求分析、设计、编码、测试等各阶段，表 1.1 列出了一般编程的解题步骤与从软件工程角度看这些步骤的对应关系。

表 1.1 程序开发的阶段

程序解题	软件工程	具体工作
建模型	需求分析阶段	提取问题要完成的功能；分析问题涉及的数据对象，找出数据对象之间的关系
设计	设计阶段	数据结构设计、软件的结构设计、算法设计
编程	编码阶段	编写程序代码
验证	测试	软件测试与调试

【知识 ABC】软件工程

软件工程是研究和应用如何以系统性的、规范化的、可定量的过程化方法去开发和维护软件，以及如何把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来。

软件工程是一门交叉性的工程学科，它是将计算机科学、数学、工程学和管理学等基本原理应用于软件的开发与维护中，其重点在于大型软件的分析与评价、规格说明、设计和演化，同时涉及管理、质量、创新、标准、个人技能、团队协作和专业实践等。从这个意义上看，软件工程可以看作由下列三部分组成：

- 计算机科学和数学用于构造软件的模型与算法；
- 工程科学用于制定规范、设计范型、评估成本以及确定权衡等；
- 管理科学用于计划、资源、质量、成本等管理。

从软件工程的角度看，软件的产生直到报废的生命周期有下面几个阶段。

（1）问题的定义及规划阶段

软件开发方与需求方共同讨论，主要确定软件的开发目标及其可行性。

（2）需求分析阶段

在确定软件开发可行的情况下，对软件系统需要实现的各个功能进行详细分析。

具体的工作是描述新系统的目的、范围、定义和功能。需求分析是软件工程中的一个关键过程。在这个过程中，系统分析员和软件工程师确定客户的需要。只有在确定了这些需要后，他们才能够分析和寻求新系统的解决方法。

在软件工程的历史中，很长时间里人们一直认为需求分析是整个软件工程中最简单的一个步骤，但在过去十年中越来越多的人认识到它是整个过程中最关键的一个过程。假如在需求分析时分析者未能正确地认识到客户的需要，那么最后的软件实际上不可能达到客户的需要，或者软件无法在规定的时间里完工。图 1.2 以漫画的形式说明了在需求的描述和理解上的困难情形。

（3）软件设计阶段

根据需求分析的结果，对整个软件系统进行设计，如数据结构设计、系统架构设计（以面向过程为例，有模块划分、模块通信、模块流程等设计）、数据库设计等。软件设计一般分为总体设计和详细设计。好的软件设计将为软件程序编写打下良好的基础。

面向过程的系统架构设计的基本过程是，按自顶向下、逐步求精的方法把系统逐层分解成各级模块，再分别设计模块通信方式和模块流程。模块划分原则是“高内聚低耦合”，即让模块具有独立功能而且和其他模块之间没有过多的相互作用。按这样的原则划分出的模块，