iCourse·教材

“高等学校本科计算机类专业应用型人才培养研究”项目规划教材

数据结构与算法

Data Structures and Algorithms

主　编　王曙燕
副主编　王春梅

高等教育出版社

iCourse·教材

“高等学校本科计算机类专业应用型人才培养研究”项目规划教材

数据结构与算法

Data Structures and Algorithms

主　编　王曙燕
副主编　王春梅
王曙燕　王春梅　初建玮　王　燕　编著

高等教育出版社·北京

内容提要

数据结构与算法设计是高校理工科类专业计算机应用能力提高的重要技术基础，也是计算机类专业的核心课程和硕士研究生入学的统考科目。本书将数据结构和算法分析与设计的基础知识相结合，以实际应用为驱动，将各种数据结构的知识融入实际问题的解决中，对相关算法的核心思想进行深入剖析，并总结比较各类算法的特点和适用范围，重点培养学生使用数据结构知识分析问题和解决问题的能力，为后续课程的学习以及从事软件开发工作打下良好的基础。

本书系统地讲解数据结构与算法设计的相关知识。全书共4篇，包含数据结构的基本概念、线性表、栈和队列、串、多维数组和广义表、树、图、查找、排序、综合实验要求及典型案例分析等内容，每章后都附有丰富的习题。为了让读者能够方便地进行网络学习，相应的“数据结构与算法”MOOC已在中国大学MOOC上线，读者可结合MOOC与教材学习本课程。随书还提供配套的教学微视频、电子教案、算法源码等课程资源。

本书既可以作为高校理工类专业“数据结构与算法”课程的教材，也可供准备考研的读者阅读参考，同时也可作为工程技术人员和计算机爱好者的参考资料。

图书在版编目(CIP)数据

数据结构与算法／王曙燕主编. --北京：高等教育出版社，2019.8

ISBN 978-7-04-052437-6

Ⅰ.①数… Ⅱ.①王… 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(2019)第159252号

策划编辑	倪文慧	责任编辑	倪文慧	封面设计	张 志	版式设计	张 杰
插图绘制	于 博	责任校对	李大鹏	责任印制	刘思涵		

出版发行	高等教育出版社	网 址	http://www.hep.edu.cn
社 址	北京市西城区德外大街4号		http://www.hep.com.cn
邮政编码	100120	网上订购	http://www.hepmall.com.cn
印 刷	河北鹏盛贤印刷有限公司		http://www.hepmall.com
开 本	850mm×1168mm 1/16		http://www.hepmall.cn
印 张	25		
字 数	560千字	版 次	2019年8月第1版
购书热线	010-58581118	印 次	2019年8月第1次印刷
咨询电话	400-810-0598	定 价	45.00元

物 料 号 52437-00

数据结构与算法

主　编　王曙燕
副主编　王春梅

1　计算机访问http://abook.hep.com.cn/1871825，或手机扫描二维码、下载并安装Abook应用。

2　注册并登录，进入“我的课程”。

3　输入封底数字课程账号（20位密码，刮开涂层可见），或通过Abook应用扫描封底数字课程账号二维码，完成课程绑定。

4　单击“进入课程”按钮，开始本数字课程的学习。

课程绑定后一年为数字课程使用有效期。受硬件限制，部分内容无法在手机端显示，请按提示通过计算机访问学习。

如有使用问题，请发邮件至abook@hep.com.cn。

http://abook.hep.com.cn/1871825

前　言

“数据结构与算法”课程是一门面向设计,且处于计算机学科核心地位的技术基础和主干必修课,也是算法分析与设计、操作系统、编译技术、计算机图形与图像处理等专业课的先修课程。国内外许多科技人员对学习、研究数据结构都非常重视。美国 IEEE 和 ACM 的教学计划把“数据结构与算法”列入计算机以及信息技术相关学科专业的本科必修基础课程及计算机核心课程之首,教育部 2018 年颁布的计算机类专业教学质量国家标准也将该课程列为专业类核心课程。

本书根据学科发展,结合教学需要对课程内容进行必要的筛选、补充、更新和重组,注重学生计算思维能力和创新实践能力的培养,并补充了后续课程和相关领域应用的实例。为了顺应互联网+教育的时代要求,秉承“以学为中心”的教育理念,方便读者进行网络自主学习,本书相应的慕课已在中国大学 MOOC 上线,并提供有课程内容及部分实验解析的微视频、电子教案、算法源码等丰富的课程资源。

本书将数据结构知识和算法分析与设计的基础知识相结合,以实际应用案例为驱动,将各种数据结构与算法的知识融入实际问题的解决中,对相关算法的核心思想进行深入剖析,并总结比较各类算法的特点和适用范围,重点培养学生利用数据结构知识分析和解决实际问题的能力,为其学习后续课程以及日后从事软件开发工作打下良好的基础。

本书的参考学时为 64~80 学时,建议采用理论实践一体化的教学模式。如果有条件,还可以配有 1~2 周的“数据结构与算法”综合课程设计。

全书共 4 篇。第一篇线性结构(第 2 章~第 5 章),主要包括线性表、栈和队列、串、多维数组和广义表;第二篇非线性结构(第 6 章、第 7 章),包括树和图结构;第三篇相关技术(第 8 章、第 9 章),主要包括查找以及排序;第四篇综合实践(第 10 章),主要包括实验题目及要求、实验报告格式、典型案例解析等内容。

本书由王曙燕任主编,王春梅任副主编。各章节编写分工如下:王曙燕编写了第 1、2、3 章,王春梅编写了第 5、7、8、9 章,初建玮编写了第 6 章,王燕编写了第 4、10 章,王曙燕和王春梅对全书进行了细致的修改和统稿。

尽管本书经过作者的反复讨论和推敲,但因水平和经验有限,书中难免有欠妥之处,恳请读者批评指正。

作者

2019 年 5 月

目　录

第一篇　线 性 结 构

第二篇 非线性结构

第三篇 相关技术

第四篇　综 合 实 践

第1章 引 言

1946年,世界上第一台电子数字式计算机在美国宾夕法尼亚大学诞生了,即电子数值积分计算机(electronic numerical integrator and calculator,ENIAC)。ENIAC奠定了电子计算机的发展基础,开辟了计算机科学技术的新纪元。在此后短短的几十年间,计算机的发展突飞猛进。伴随着硬件的发展,计算机软件、计算机网络技术也得到了迅猛发展,计算机的应用领域也由最初的数值计算扩展到人类生活的各个领域。计算机已成为最基本的信息处理工具,广泛应用于办公自动化、企事业管理与决策、工程设计、气象预报、火箭发射等信息管理、科学计算和过程控制等领域。当前大数据、云计算、人工智能技术蓬勃发展,已形成规模巨大的计算机产业,带动了全球范围的技术进步,计算机的发展已进入一个快速而又崭新的时代,计算机技术正朝着巨型化、微型化、网络化和智能化方向发展。

微视频 1-1:数据结构的基本概念-1

早期的计算机主要用于数值计算,随着计算机科学的发展,计算机的应用已远远超出了单纯进行数值计算的范围,数据处理已在计算机应用中占有越来越重要的地位。据统计,目前非数值型信息(如文字、字符、图、树、表、视频、音频等)的处理占据了计算机90%以上的机时。如何有效地组织数据,以便设计出更高效与高质量的程序来解决现实中的问题,已成为计算机科学工作者十分关心的事情。

"数据结构"作为一门独立的课程,最早由美国的一些大学开设。1968年美国计算机科学家Donald E. Knuth创立了数据结构的最初体系,他所著的《计算机程序设计技巧 第一卷 基本算法》是第一本系统阐述数据的逻辑结构、存储结构及其操作的著作。20世纪60年代末~70年代初出现了大型程序,软件也相对独立,结构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结构,加上设计一种好的算法。20世纪70年代中期~80年代初,各种版本的数据结构著作相继出现。目前在我国,"数据结构"也已不仅仅是计算机专业的核心课程之一,还是其他理工类专业的主要选修课程之一。

著名的计算机科学家Niklaus Wirth曾提出一个著名的公式:

数据结构+算法=程序

该公式阐明了数据结构和算法对于程序设计的重要性。数据结构与算法的研究涉及计算机求解问题过程的两个重要方面:刻画实际问题中信息及其关系的数据结构和描述问题解决方案的逻辑抽象算法。

用计算机解决实际问题时,首先要确定其数据结构,然后再确定采用的算法策略。程序设计

则是为计算机编制处理问题的指令的一组集合。

对一个实际的问题，要用计算机来编程和实现，首先要做的工作就是数据抽象。当抽象出问题的数学模型之后，再来编写程序，问题就可以迎刃而解。对于大多数数值计算的程序问题，比如物理学上的结构静力分析计算，还有全球天气预报的计算等问题，可以用线性代数的方程组或环流模式的方程来解决，列出方程组之后再编程就是很容易的事情，也就是说计算机编程对于数值型数据问题相对比较好处理。

微视频 1-2：数据结构的基本概念-2

但是很多计算机的应用是非数值型计算问题，该类问题编程时要解决的首要问题就是其表示和存储。例如，要编一个下象棋的程序，首先要解决棋盘状态数据在计算机中的存储问题，即用什么样的数据结构进行存储。

其次就是它的操作，即如何对这些数据对象进行处理和访问。这两个问题如果解决了，用计算机解决问题的方法也就找到了。

注：在微视频“数据结构的基本概念”中，通过分析超市商品问题、人机对弈问题和多岔路口交通灯的管理问题引出了数据结构的概念，读者可以扫码自行观看。

1.1　数据结构的概念

数据结构主要研究数据（特别是非数值型数据）的组织、存储及运算方法。本节首先介绍数据结构的相关术语及概念。

1. 数据

数据（data）是描述客观事物的数值、字符以及能输入计算机且能被处理的各种符号集合。换句话说，数据是对客观事物采用计算机能够识别、存储和处理的形式所进行的描述。数据不仅包括整型、实型、布尔型等数值型数据，还包括字符、声音、图像等符号集合。例如，一个编译程序或文字处理程序的处理对象是字符串。

2. 数据元素

数据元素（data element）是组成数据的基本单位，是数据集合的个体，在计算机中通常作为一个整体进行考虑和处理。一个数据元素可由一个或多个数据项组成，**数据项（data item）是数据不可分割的最小单位**。例如，通讯录是数据，其中每一个人的信息就是一个数据元素，通常称为“记录”（record）；而通讯录中的每一项（如姓名、电话号码等）即为一个数据项，如表 1-1 所示。

3. 数据对象

数据对象（data object）是性质相同的数据元素的集合，是数据的一个子集。例如，正整数数据对象是集合 $N=\{0,1,2,\cdots\}$，字母字符数据对象是集合 $C=\{'A','B',\cdots,'Z'\}$，表 1-1 中的通讯录也可看作一个数据对象。

表 1-1 通 讯 录

序号	姓名	性别	电话号码	单位	地址
1	王平	女	13008892088	西安交通大学	西安市友谊东路
2	李明	男	13992900001	西安邮电大学	西安市长安区
3	张鹏飞	男	13200022223	西安邮电大学	西安市长安区
…	…	…	…	…	…

4. 数据结构

数据结构(data structure)是指相互之间存在一种或多种特定关系的数据元素集合。通常数据对象中的数据元素不是孤立的,而是彼此之间存在某种关系,例如,表结构(表 1-1,元素之间存在线性关系),树形结构(图 1-1,元素之间存在一对多的层次关系),图结构(图 1-2,元素之间存在多对多的任意关系)。数据元素相互之间的关系称为“结构”,即数据的组织形式,所以也可以说数据结构是带有结构的数据元素的集合。

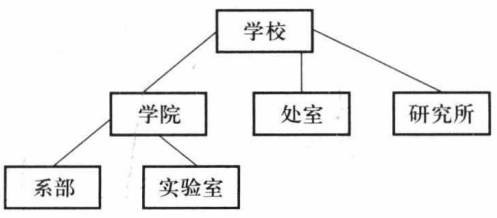

图 1-1 树形结构示例

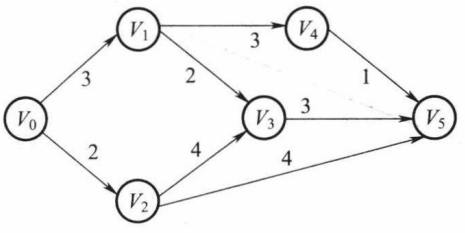

图 1-2 图结构示例

数据结构是一个二元组: Data_Structure=(D,R)。其中 D 是数据元素的有限集,R 是 D 上关系的有限集。

5. 数据类型

数据类型(data type)是一组性质相同的值集合,以及定义在这个值集合上的一组操作的总称。数据类型中定义了两个集合,即该类型的取值范围和该类型中可允许使用的一组运算。例如,高级语言中的数据类型就是已经实现的数据结构的实例。数据类型是高级语言中允许的变量种类,如在高级语言中的整型类型可能的取值范围是某个区间(区间大小和不同的机器相关),可进行的运算为加、减、乘、除、乘方、取余(如 C 语言中的+、-、*、/、%)。

从硬件的角度看,它们的实现涉及“字”“字节”“位”“位运算”等;从用户的角度看,他们并不需要了解整数在计算机内是如何表示、运算细节是如何实现的,而仅需要了解整数运算的外部运算特性,不必知道机器内部位运算的细节就可运用高级语言进行程序设计。引入数据类型的目的是实现信息隐藏,将一切用户不必关心的细节封装在类型中。如两整数求和问题,程序用户只需注重其数学求和的抽象特性,无需关心加法运算涉及的内部位运算实现。

按“值”的不同特性，高级程序语言中的数据类型可分为两大类：一类是非结构的原子类型，其值是不可分解的，如C语言中的标准类型（整型、实型、字符型等）及指针；另一类是结构类型，其值是由若干成分按某种结构组成的，因此是可以分解的，并且其成分可以是非结构的，也可以是结构的，如数组、结构体等。

6. 数据抽象与抽象数据类型

抽象是对一种事物或一个系统的简化描述，它关注事物或系统的本质方面，而忽略非本质的细节。

计算机科学家E. W. Dijkstra在谈到关于如何应对日益增长的软件复杂度问题时说，设计并实现一个大规模软件的中心问题是怎样减小复杂度，一个途径就是通过抽象。软件技术发展的历史证明了这一点。程序设计语言从机器语言→汇编语言→高级语言→非过程化语言（面向对象语言）的发展就是不断抽象的过程。因为用户关心的只是软件能做什么，而不是为什么能这样做及其具体的实现细节。

前面讨论的数据类型抽象程度不够高，且对问题来说，不能比较自然地加以表达，因为它只描述了数据的形式，而没有描述它们的功能。

（1）抽象数据类型

抽象数据类型（abstract data type，ADT）是指一个数学模型以及定义在该模型上的一组操作，或者说是基于一类逻辑关系的数据类型以及定义在该类型之上的一组操作。“抽象”的意义在于数据类型的数学抽象特性。抽象数据类型的定义取决于客观存在的一组逻辑特性，而与其在计算机内如何表示和实现无关，即不论其内部结构如何变化，只要它的数学特性不变，都不影响其外部使用。在某种意义上讲，抽象数据类型和数据类型实质上是一个概念。整数类型就是一个简单的抽象数据类型实例，尽管在不同处理器上实现的方法可以不同，但其定义的数学特性是相同的，都可以实现+、-、*、/、%等运算，在用户看来都是相同的。

一个抽象数据类型定义了一个数据对象、数据对象中各元素间的结构关系以及一组处理数据的操作。抽象数据类型通常是由用户定义，用以表示应用问题的数学模型，它由基本的数据类型组成，并包括一组相关的服务操作。

抽象数据类型包括定义和实现两方面，其中定义是独立于实现的。定义仅给出一个ADT的逻辑特性，不必考虑其如何在计算机中实现。抽象数据类型的特征是使用与实现分离，实现封装和信息隐藏。就是说，在设计抽象数据类型时，类型的定义与其实现分离。

抽象数据类型的范畴包括各种不同的计算机处理器中已定义并实现的固有数据类型，还包括用户在设计软件系统时自己定义的数据类型。所定义的数据类型的抽象层次越高，含有该抽象数据类型的软件复用程度就越高。

抽象数据类型的物理实现作为私有部分封装在其实现模块内，使用者无法看到，也不能直接操作该类型所存储的数据，只能通过界面中的服务来访问这些数据。从实现者的角度来看，把抽象数据类型的物理实现封装起来，既有利于编码与测试，也有利于修改。在改进数据结构时，只要界面服务的使用方式不变，则只需改变抽象数据类型的物理实现，所有使用该抽象数据类型的程序均不需要改变，提高了系统的稳定性。

抽象数据类型是近年来计算机科学中提出的最重要的概念之一，它集中体现了程序设计中一些最基本的原则：分解、抽象和信息隐藏。

一个抽象数据类型确定了一个模型，但将模型的实现细节隐藏起来；它定义了一组运算，但将运算的实现过程隐藏起来。可以用抽象数据类型来指导问题求解的过程。

（2）抽象数据类型的定义格式

抽象数据类型的定义格式如下。

```
ADT <ADT 名>
{ 数据对象:<数据对象的定义>
  结构关系:<结构关系的定义>
  基本操作:<基本操作的定义>
}ADT <ADT 名>
```

其中，数据对象和结构关系的定义采用数学符号和自然语言描述，而基本操作的定义格式如下。

```
<操作名称>（参数表）
操作前提:<操作前提描述>
操作结果:<操作结果描述>
```

例如，一个线性表的抽象数据类型描述如下。

```
ADT Linear_list
{
  数据对象:所有 a_i 属于同一数据对象，i=1,2,…,n   n≥0;
  结构关系:所有数据元素 a_i(i=1,2,…,n-1)存在次序关系<a_i,a_i+1>，a_1无前驱，a_n无后继;
  基本操作:设 L 为 Linear_list;
          Initial(L):初始化空线性表;
          Length(L):求线性表表长;
          Get(L,i):取线性表的第 i 个元素;
          Insert(L,i,b):在线性表的第 i 个位置插入元素 b;
          Delete(L,i):删除线性表的第 i 个元素;
}ADT Linear_list
```

上述抽象数据类型很明显是抽象的。数据元素所属的数据对象没有局限于一个具体的整型、实型或其他类型。所具有的操作也是抽象的数学特性，并没有具体到何种计算机语言指令与程序编码。

（3）抽象数据类型的实现

抽象数据类型的实现需要借助于高级语言，并且其具体实现也依赖于所选择的高级语言的功能。从程序设计的历史发展来看，有传统的面向过程的程序设计、“包”“模型”的设计方法、面向对象的程序设计等不同的实现方法。

下面分三种情况予以介绍。

① **传统的面向过程的程序设计**。也就是现在常用的方法，根据逻辑结构选定合适的存储结构，根据所要求的操作设计出相应的子程序或子函数。

在标准 Pascal、C 等面向过程的语言中，用户可以自己定义数据类型。由此可以借助过程和函数、利用固有的数据类型来表示和实现抽象数据类型。由于标准 Pascal 语言的程序结构框架是由严格规定次序的“段”（包括程序首部、标号说明、常量定义、类型定义、变量说明、过程或函数说明和语句部分）组成，因此，所有使用已定义的抽象数据类型的外部用户必须将这些抽象数据类型说明和过程说明嵌入自己程序的适当位置。

例如，用 C 语言实现抽象数据类型时，数据类型可以用基本数据类型、结构体类型，也可用 typedef 自定义类型，以增强抽象性和可读性；基本操作可以用 C 语言的子函数实现。

② **“包”“模型”的设计方法**。Ada 语言提供了包（package），Module-2 语言提供了模块（module）结构，Turbo Pascal 语言提供了单元（unit）结构。每个模块可含有一个或多个抽象数据类型，它不仅可以单独编译，而且为外部使用抽象数据类型提供了方便，用这类结构实现抽象数据类型比起第一种方法有一定的进步。

③ **面向对象的程序设计**（Object Oriented Programming，OOP）。在面向对象的程序设计语言中，借助对象描述抽象数据类型，存储结构和操作函数的说明被封装在一个整体结构中，这个整体结构称为“类”（class）。属于某个类的具体变量称为“对象”（object）。面向对象的程序设计与抽象数据类型的实现更加接近和一致。从前面对数据类型的讨论中可以看到，在面向对象的程序设计语言中，“类型”的概念与“操作”密切相关，同一种数据结构和不同的操作组合将构成不同的数据类型，结构说明和过程说明被统一在一个整体对象之中。其中，数据结构的定义为对象的属性域，过程或函数的定义在对象中称为“方法”（method），是对象的性能描述。

面向对象的开发方法首先着眼于应用问题所涉及的对象，包括对象、对象属性和要求的操作，借助对象描述抽象数据类型，存储结构和操作函数的说明被封装在类中，软件易于修改。而传统的结构化的开发方法是面向过程的开发方法，首先着眼于系统要实现的功能。

1.2　数据结构的内容

微视频 1-3：数据的逻辑结构和存储结构

“数据结构”课程在计算机类专业中是一门综合性的专业基础课。数据结构的研究不仅涉及计算机硬件（特别是编码理论、存储装置和存取方法等）的范畴，而且和计算机软件的研究有着更密切的关系，无论是编译程序还是操作系统，都涉及数据元素在存储器中的分配问题。在研究信息检索时也必须考虑如何组织数据，以使查找和存取数据元素更为方便。因此，可以认为该课程是介于数学、计算机硬件和计算机软件三者之间的一门核心课程。在计算机科学中，数据结构不仅是一般程序设计（特别是非数值计算的程序设计）的基础，还是设计和实现编译程序、操作系统、数据库系

统及其他系统程序和大型应用程序的重要基础。它包含三方面的内容:数据的逻辑结构、数据的存储结构和数据的运算。

1.2.1 数据的逻辑结构

数据的逻辑结构是指数据元素之间逻辑关系的描述。

可以用一个二元组给出其形式化的描述:

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

其中 D 是数据元素的有限集,R 是 D 上关系的有限集。

这里的关系描述的是数据元素之间的逻辑关系,因此又称为数据的“逻辑结构”。一个数据元素通常称为一个“结点”,描绘时用一个圆圈表示。根据数据元素之间关系的不同特性,通常有 4 种基本结构:集合结构、线性结构、树形结构和图状结构,如图 1-3 所示。

图 1-3 4 类基本结构关系图

① 集合结构:结构中的数据元素之间除了同属于一个集合的关系外,无任何其他关系。

② 线性结构:结构中的数据元素之间存在着一对一的线性关系。

③ 树形结构:结构中的数据元素之间存在着一对多的层次关系。

④ 图状结构:结构中的数据元素之间存在着多对多的任意关系。

根据数据元素之间关系的不同特性,数据结构又可分为两大类:线性结构和非线性结构。按照这种划分原则,本书将要介绍的数据结构划分为线性结构(线性表、栈、队列、字符串、数组和广义表),非线性结构(树和图),此外还包括介绍基本的数据处理技术——查找和排序。

1.2.2 数据的存储结构

数据的逻辑结构是从逻辑上来描述数据元素之间的关系,是独立于计算机的。然而讨论数据结构的目的是在计算机中实现对它的操作,因此还需要研究数据元素和数据元素之间的关系如何在计算机中表示,这就是数据的存储结构。

存储结构是逻辑关系的映像与元素本身的映像,是数据结构的实现;逻辑结构是数据结构的抽象。

计算机的存储器由很多存储单元组成,每个存储单元有唯一的地址。数据存储结构要讨论的就是数据结构在计算机存储器上的存储映像方法。数据结构在计算机中的表示(又称映像)称为数据的“物理结构”或“存储结构”。它包括数据元素的表示和关系的表示。

数据元素在计算机中用若干个二进制“位串”表示。

数据元素之间的关系在计算机中有两种表示方法:顺序映像和非顺序映像,由此可得到两种不同的存储结构:顺序存储和链式存储。顺序存储的特点是借助元素在存储器中的相对位置来表示数据元素之间的逻辑关系;链式存储的特点是借助指针表示数据元素之间的逻辑关系。

任何一个算法的设计(决定有什么样的操作或运算)取决于选定的数据(逻辑)结构,而算法的实现则依赖于采用的存储结构。

学习数据结构的目的,不仅是了解数据之间的逻辑结构以及它们在计算机中的存储结构,更重要的是实现在此基础上进行的各种增、查、改、删等基本运算。因此,在逻辑和存储结构上的运算集合也是数据结构要讨论的重要内容之一。

综上所述,数据结构的内容可归纳为三部分,逻辑结构、存储结构和运算集合。按某种逻辑关系组织起来的一批数据,按一定的映像方式把它存放在计算机存储器中,并在这些数据上定义了一个运算的集合,就叫做"数据结构"。

1.3　算　　法

微视频 1-4:算法及其时间复杂度

开发程序的目的,就是要解决实际问题。然而,面对各种复杂的实际问题应如何编制程序,往往令初学者感到茫然。程序设计语言只是一个工具,只懂得语言的规则并不能保证编制出高质量的程序。程序设计的关键是设计算法,算法与程序设计和数据结构密切相关。简单地讲,算法是解决问题的策略、规则和方法。算法的具体描述形式很多,但计算机程序是对算法的一种精确描述,而且可在计算机上运行。

1.3.1　算法的概念

算法就是解决问题的一系列操作步骤的集合。比如,厨师做菜时,都要经过一系列的步骤:洗菜、切菜、配菜、炒菜和装盘。用计算机解题的步骤即为算法,编程人员必须告诉计算机先做什么、再做什么,并通过高级语言的语句来实现。通过这些语句,一方面体现了算法的思想,另一方面指示计算机按算法的思想去工作,从而解决实际问题。程序就是由一系列语句组成的。

在前面提到的 Niklaus Wirth 的公式"数据结构+算法=程序"中,数据结构是指对数据(操作对象)的描述,即数据的类型和组织形式,算法则是对操作步骤的描述。也就是说,数据描述和操作描述是程序设计的两项主要内容。数据描述的主要内容是基本数据类型的组织和定义,数据操作则是由语句来实现的。

算法具有下列特性。

① 有穷性。对于任意一组合法输入值,在执行有穷步骤之后一定能结束,即算法中的每个步骤都能在有限时间内完成。

② 确定性。算法的每一步必须是确切定义的,使算法的执行者或阅读者都能明确其含义及