

计算机 算法理论与应用

蒋伟 田嵩 孙鹏◎著

JISUANJI
SUANFA LILUN
YU YINGYONG



中国水利水电出版社
www.waterpub.com.cn

计算机算法理论与应用

蒋伟 田嵩 孙鹏 著



中国水利水电出版社
www.waterpub.com.cn

· 北京 ·

内 容 提 要

计算机在解题的过程中,无论是形成解题思路还是编写程序,都是在实施某种算法,本书对计算机算法的理论与应用进行了深入分析,首先阐述了计算机算法基础知识,而后分别论述了计算机神经网络算法与应用、数据挖掘算法与应用、MATLAB 算法与应用、工程图形算法与应用、数字视频图像处理算法与应用、智能算法与应用、蚁群算法及其应用、群体智能算法及其应用、高维多目标进化算法与应用及视觉算法在智能车中的应用,最后分析了网络环境下计算机的安全性问题。

本书可供计算机科学等相关专业本科生、研究生阅读,也可供从事计算机算法设计、分析与应用工作的教师与研究人员参考。

图书在版编目 (CIP) 数据

计算机算法理论与应用 / 蒋伟, 田嵩, 孙鹏著. --

北京: 中国水利水电出版社, 2018. 7

ISBN 978-7-5170-6582-1

I. ①计… II. ①蒋… ②田… ③孙… III. ①计算机
算法 IV. ①TP301.6

中国版本图书馆 CIP 数据核字(2018)第 128195 号

责任编辑: 陈 洁

封面设计: 王 伟

书 名	计算机算法理论与应用 JISUANJI SUANFA LILUN YU YINGYONG
作 者	蒋伟 田嵩 孙鹏 著
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址: www. waterpub. com. cn E-mail: mchannel@ 263. net (万水) sales@ waterpub. com. cn 电话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	三河市同力彩印有限公司
规 格	185mm×260mm 16 开本 12 印张 292 千字
版 次	2018 年 9 月第 1 版 2018 年 9 月第 1 次印刷
印 数	0001—2000 册
定 价	48. 00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

前言

电子计算机 (electronic computer) 是一种能自动地、高速地进行大量运算的电子设备。它能通过对输入的数据进行指定的数值运算和逻辑运算来求解各种算题, 也能用来处理各种数据和事务, 是一种自动化信息处理工具。当它与一定的机电设备或仪器设备相结合时, 能实现对生产过程和实验过程的控制。

严格地说, 电子计算机本身就是近代数学的辉煌成就, 将计算机与数学割裂开来, 既不合理也不可能。组合学也就是在计算机科学蓬勃发展的刺激下而崛起的, 从而成为比较活跃的数学分支。组合学从与计算机科学相结合中获得了广阔的发展空间, 从而也为计算机科学奠定了理论基础。

那么, 计算机在解题的过程中, 无论是形成解题思路还是编写程序, 都是在实施某种算法。计算机算法是计算机科学和计算机应用的核心, 无论是计算机系统、系统软件的设计, 还是为解决计算机的各种应用课题做的设计都可归结为算法的设计。

本书对计算机算法的理论与应用进行了深入分析, 首先阐述了计算机算法基础知识, 而后分别论述了计算机神经网络算法与应用、数据挖掘算法与应用、MATLAB 算法与应用、工程图形算法与应用、数字视频图像处理算法与应用、智能算法与应用、蚁群算法及其应用。

随着计算机技术的迅速发展, 人们对计算机的应用要求在质和量上也在不断提高。传统的单机运行方式已无法适应信息社会的需求, 现代社会是信息社会, 网络对人们的学习、工作和生活以及对社会的影响越来越大, 使得人们都希望掌握一定的网络知识。计算机网络的开发、研究以及培养该领域的核心人才越来越受到全社会的广泛关注。计算机网络应用的迅速普及, 要求人们了解、掌握网络的各种技术, 以适应信息社会发展的需要。

在计算机网络深入普及的信息时代, 信息本身就是时间, 就是财富。信息通过脆弱的公共信道传输, 储存于“不设防”的计算机系统中。如何保护信息的安全使之不被窃取及不至于被篡改或破坏, 已成为当今普遍关注的问题。密码是有效而且可行的办法, 在计算机网络的刺激下, 近代密码学便在算法复杂性理论的基础上建立起来了, 密码最早是应

用于战争当中的，后来随着人们对信息安全的重视，密码逐渐的转移到了保护人们信息安全的用途上来。事实上，资源共享和网络安全是一对矛盾，随着资源共享的加强，网络信息安全问题也日益突出。对此种现状，本书对计算机网络安全体系、技术及安全等方面的问题也进行了论述。

本书在写作过程中得到了相关领导的支持和鼓励，同时参考和借鉴了有关专家、学者的研究成果，在此表示诚挚的感谢！由于时间及能力有限，书中难免存在疏漏与不妥之处，欢迎广大读者给予批评指正！

目 录

前言

第一章 计算机算法概述.....	1
第一节 计算机算法基础知识.....	1
第二节 算法复杂性分析	10
第三节 问题复杂度与算法复杂度关系探索	20
第二章 计算机神经网络算法与应用	22
第一节 神经网络概述	22
第二节 神经网络感知机	28
第三节 神经网络在天气预测、疾病预测及客户聚类中的应用.....	34
第三章 计算机数据挖掘算法与应用	44
第一节 数据挖掘概述	44
第二节 数据挖掘及其在电子商务中的应用	53
第四章 计算机 MATLAB 算法与应用	64
第一节 MATLAB 常用算法分析	64
第二节 MATLAB 算法在金融预测中的应用	71
第五章 计算机工程图形算法与应用	88
第一节 计算机辅助图形算法概述	88
第二节 工程曲线程序设计与绘制	96
第三节 工程曲面程序设计与绘制.....	103
第四节 曲面交线与展开图的计算机图形程序设计与应用.....	108
第六章 计算机数字视频图像处理算法与应用.....	113
第一节 数字视频图像处理算法.....	113
第二节 数字视频图像处理在视频监控系统中的应用.....	120

第七章 计算机智能算法与应用.....	133
第一节 混沌算法及其在数字图像加密中的应用.....	133
第二节 粒子群优化算法及其应用.....	141
第三节 多 Agent 算法及其在故障诊断中的应用	146
第八章 计算机蚁群算法及其应用.....	157
第一节 蚁群算法概述.....	157
第二节 蚁群算法的改进.....	164
第三节 蚁群算法在实际优化问题中的应用.....	170
参考文献.....	177

第一章 计算机算法概述

计算机算法是以一步一步的方式来详细描述计算机如何将输入转化为所要求的输出的过程。或者说，算法是对计算机上执行的计算过程的具体描述。

第一节 计算机算法基础知识

一、算法的概念

算法，简言之就是解决问题的方法。人们解决问题的过程一般由若干步骤组成，通常把解决问题的确定方法和有限步骤称为算法。如果相关问题的解决最终由计算机来实现，又由于计算机不具备思考能力以及人的“跳跃性思维”等因素，因此方法的确定和对步骤的描述尤为重要。

算法是对解题过程的描述，这种描述是建立在程序设计语言这个平台之上的。就算法的实现平台而言，可以抽象地对算法作如下定义：

算法=控制结构+原操作（对固有数据类型的操作）

无论是面向对象的程序设计语言，还是面向过程的程序设计语言，都是用三种基本结构（顺序结构、选择结构和循环结构）来控制算法流程的。每个结构都应该是单入口单出口的结构体。结构化算法设计常采用自顶向下逐步求精的设计方法，因此，要描述算法首先需要有表示三个基本结构的构件，其次能方便支持自顶向下逐步求精的设计方法。

二、算法与数据结构及程序的关系

（一）算法与数据结构的关系

算法与数据结构是密不可分的。除极少数算法外，几乎所有算法都需要数据结构的支持，而且数据结构的优劣往往决定算法的好坏。数据结构把输入的数据及运算过程中产生的中间数据以某种方式组织起来以便于动态地寻找、更改、插入、删除等。没有一种数据结构是万能的，我们应根据问题和算法的需要选用和设计数据结构，而在讨论数据结构时也必定会讨论其适用的算法。所以，数据结构课程与算法课程的内容往往有很大重叠。但是，数据结构课程需要解释其在计算机上的具体实现，而算法课程着重讨论在更为抽象的

层次上解决问题的技巧及分析方法。打个比方,数据结构就好像汽车零件,例如发动机、车轮、车窗、车闸、座椅、灯光、方向盘等,而算法就好像是汽车总体设计。我们假定读者熟悉常用的一些数据结构,包括数组、队列、堆栈、二叉树等,而略去对它们的介绍。读者还应当具有基本的编写程序的知识。

(二) 算法与程序的关系

一段用某种计算机语言写成的源码,如果可以在计算机上运行并正确地解决一个问题,则称为一个程序。程序必须严格遵守该语言规定的语法(包括标点符号),并且编程时往往还必须考虑到计算机的物理限制(例如,最大允许的整数在32位机上和16位机上是不同的),而算法则不依赖于某种语言,更不依赖具体计算机的限制。只要步骤和逻辑正确,一个算法可以用任何一种语言表达。当然这种语言必须清楚无误地定义每一步骤且能够让稍懂程序的人看懂。这样,设计算法者可以着重考虑解题方法而免去不必要的琐碎的语法细节。所以,算法通常不是程序,但一定可以用任一种语言的程序来实现(我们假定机器有足够大的内存)。

著名计算机科学家尼克劳斯·沃思(Niklaus Wirth)就此提出一个公式:

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

数据结构是对数据的描述,而算法是对运算操作的描述。

实际上,一个程序除了数据结构与算法这两个要素之外,还应包括程序设计方法。一个完整的C程序除了应用C语言对算法进行描述之外,还包括数据结构的定义以及调用头文件的指令。

如何根据案例的具体情况确定并描述算法,如何为实现该算法设置合适的数据结构,是求解实际案例必须面对的问题。

我们举一个例子——选择排序(selection sort)——来说明算法与程序的关系。排序问题就是要求把 n 个输入的数字从小到大(或从大到小)排好。我们假定这 n 个输入的数字是存放在数组 $A[1..n]$ 中。下面的算法称为选择排序。

【例1-1】选择排序。

输入: $A[1], A[2], \dots, A[n]$

输出: 把输入的 n 个数重排使得 $A[1] \leq A[2] \leq \dots \leq A[n]$

Selection-Sort($A[1..n]$)

```
1  for( $i \leftarrow 1, i \leq n, i++$ )
2       $key \leftarrow i$ 
3      for( $j \leftarrow i, j \leq n, j++$ )
4          if  $A[j] < A[key]$ 
5              then  $key \leftarrow j$ 
6          endif
7      endfor
8       $A[i] \leftrightarrow A[key]$ 
9  endfor
10 End
```

这个算法看上去像 C++ 程序, 但不是。实际上, 它不遵守目前为止任一个可在计算机上编译的语言规定的语法, 但它把算法的步骤描述得很清楚。这个算法含有 n 步, 对应于变量 i 从 1 变到 n 。第一步, 它把最小数选出并放在 $A[1]$ 中; 第二步, 它把余下的在 $A[2..n]$ 中的最小数选出并放在 $A[2]$ 中, ……; 第 i 步, 它把余下的在 $A[i..n]$ 中最小的数选出并放在 $A[i]$ 中。当 $i=n$ 时, 排序完成。算法中第 2 行到第 7 行表明该算法是用顺序比较的方法找到 $A[i..n]$ 中最小的数所在的位置 $A[key]$, 然后交换 $A[i]$ 和 $A[key]$, 该算法显然是正确的。

三、表示算法的方式

表示算法的方式主要有自然语言、流程图、盒图、PAD 图、伪代码和计算机程序设计语言。

(一) 自然语言

自然语言是人们日常所用的语言, 如汉语、英语、德语等, 使用这些语言不用专门训练, 所描述的算法自然也通俗易懂。

(二) 流程图

流程图是描述算法的常用工具。就简单算法的描述而言, 流程图优于其他描述算法的语言。

流程图的基本组件如图 1-1 所示。



图 1-1 流程图的基本组件

以下是流程图的三种基本控制结构的描述。

1. 顺序结构

流程图的顺序结构如图 1-2 所示。

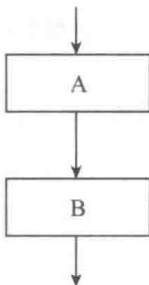


图 1-2 顺序结构

2. 选择结构

if-then-else 型分支如图 1-3 所示；do-case 型多分支如图 1-4 所示。

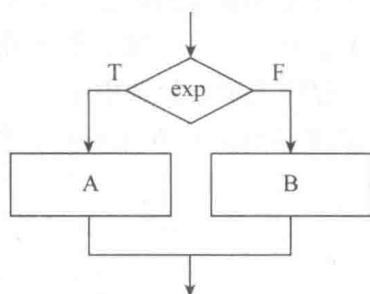


图 1-3 双分支选择结构

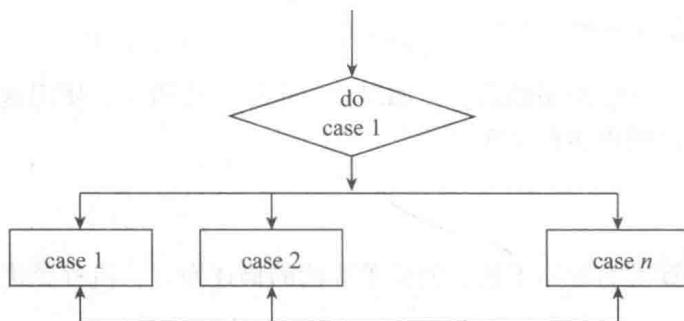


图 1-4 多分支选择结构

3. 循环结构

do-while 型循环如图 1-5 所示；do-until 循环结构如图 1-6 所示。

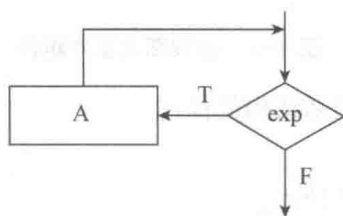


图 1-5 当型循环结构

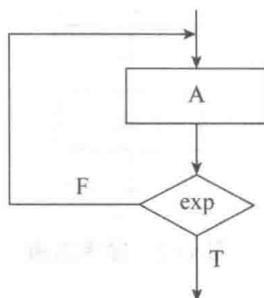


图 1-6 直到型循环结构

算法流程图虽然看起来清晰简单，但是具有一定的局限性，并没有纵观全局，所以说它并不是逐步求精的好工具。而且随意性太强，逻辑不严谨，结构化和层次感都不明显。

(三) 盒图

盒图（NS 流程图）基本组件只有三种基本控制结构，因此能强迫算法结构化。盒图的基本控制结构可分为顺序结构、选择结构及循环结构三种。

以下是盒图的三种基本控制结构的描述。

(1) 顺序结构。盒图的顺序结构如图 1-7 所示。

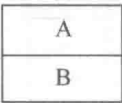


图 1-7 顺序结构

(2) 选择结构。盒图的选择结构如图 1-8 和图 1-9 所示。

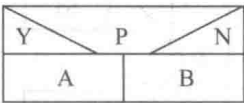


图 1-8 双分支选择结构

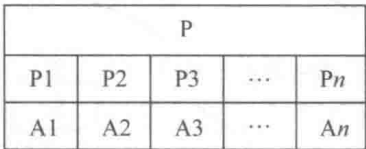


图 1-9 多分支选择结构

(3) 循环结构。盒图的循环结构如图 1-10 和图 1-11 所示。

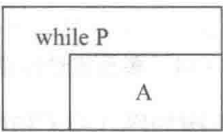


图 1-10 当型循环结构

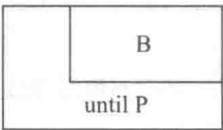


图 1-11 直到型循环结构

(四) PAD 图

问题分析图 (Problem Analysis Diagram, PAD) 是一个二维树形结构图, 层次感强、嵌套明确且有清晰的控制流程, 综合了自然语言、流程图、盒图等算法描述方式的优点。

(1) 顺序结构。PAD 图的顺序结构如图 1-12 所示。

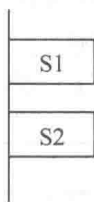


图 1-12 顺序结构

(2) 选择结构。PAD 图的选择结构如图 1-13 和图 1-14 所示。

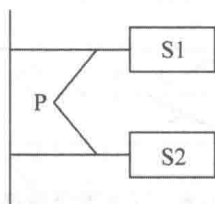


图 1-13 双分支选择结构

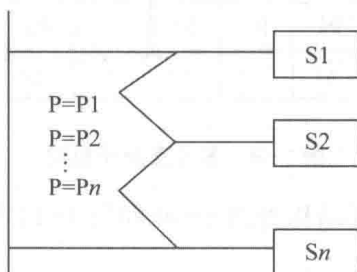


图 1-14 多分支选择结构

(3) 循环结构。PAD 图的循环结构如图 1-15 和图 1-16 所示。

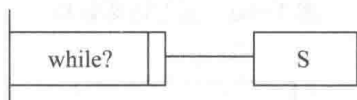


图 1-15 当型循环结构

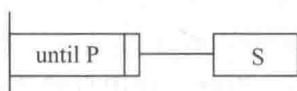


图 1-16 直到型循环结构

图 1-17 是用问题分析图描述的一个算法模块。

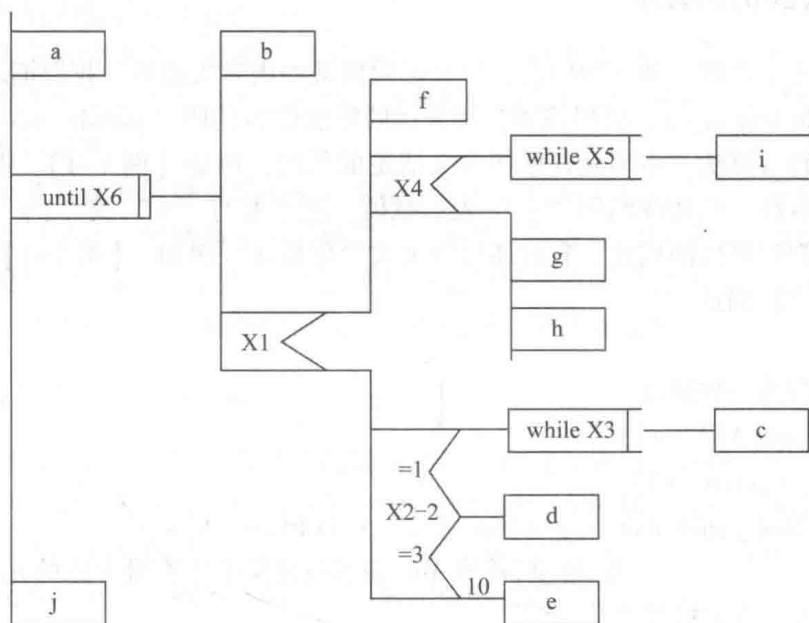


图 1-17 问题分析图实例

PAD 图的优点如下：

- (1) 使用表示结构化控制结构的 PAD 符号设计出来的算法一定是结构化的，这一点毋庸置疑。
- (2) 使用 PAD 图对算法进行描绘，结构清晰，一目了然。
- (3) 使用 PAD 图对算法进行描述，有利于用户的理解与记忆。
- (4) 很容易将 PAD 图转换成高级程序语言源程序，这种转换可由软件工具自动完成。
- (5) 不仅可以表示算法逻辑，还可以描绘数据结构。
- (6) PAD 图的符号支持自顶向下、逐步求精方法的使用。

PAD 图的缺点：由于 PAD 是用图形符号书写，与其他语言相比，编辑、录入操作不方便。

(五) 伪代码

伪代码介于自然语言和计算机语言之间，不用图形符号，可以将整个算法运行过程的结构用接近自然语言的形式描述出来，使被描述的算法可以容易地以任何一种编程语言实现。与程序设计语言相比，使用伪代码对算法进行描述，更便于理解。

(六) 计算机程序设计语言

计算机只能识别程序设计语言,因此,使用自然语言、流程图、PAD图、盒图、伪代码对算法进行描述最终还是要转换为计算机可以识别的程序设计语言。程序设计语言是一种被标准化的交流技巧,用来向计算机发出指令,具有其他语言无法比拟的严谨性。

四、算法的伪码表示

算法的描述不依赖于某一种语言,但又必须用某种语言去描述。我们把这个语言称为伪语言(pseudo language),而用该语言所表达的算法称为伪码(pseudo code)。不同的人可用不同的伪码写算法。本书允许任何含义清楚的伪码,例如【例1-1】。但是我们应当注意符号的一致性,例如我们用“ $\leftarrow$ ”表示赋值,则不要与“=”或“:=”混用。用伪码可以方便我们对算法的描述,有时还可以大大简化描述。例如,【例1-1】中的算法还可以用【例1-2】描述。

【例1-2】

选择排序的另一种描述。

Selection-Sort($A[1..n]$)

```
1.  for( $i \leftarrow 1, i \leq n, i++$ )
2.      find  $j$  such that  $A[j] = \text{Min}\{A[i], A[i+1], \dots, A[n]\}$ 
3.          % 这里,符号 Min 表示找出集合中有最小值的元素
4.       $A[i] \leftrightarrow A[j]$ 
5.  endfor
6.  End
```

显然,这样的伪码大大简化了描述,凸显了思路和方法,且易于分析。伪码应适当加入中文注释,以简洁、准确为原则。

五、基本的数据结构

(一) 线性结构

最重要最基本的数据结构是数组和链表。它们的特点是除第一个和最后一个元素外,其余的每个元素都仅有一个直接前驱和一个直接后继,这样组成了一种一对一的顺序结构。

线性表的实现方式有顺序方式和链式方式,顺序方式通常利用数组完成,链式方式通过链表实现。数组通过下标对线性结构进行随机存取,但同时带来的问题是当有元素插入或删除时将会引起大规模的数据移动。链表可以方便地解决数据插入和删除的问题,但是当访问某个元素时只能从头开始顺序查找。

数组和链表都属于一种称为线性列表的抽象数据结构,也是线性列表最主要的两种表现形式。列表是由数据项构成的有限序列,即按照一定的线性顺序排列的数据项集合。使用最多的两种特殊形式是栈和队列。栈是插入和删除操作都只能在栈顶进行的数据结构,

它的特点是后进先出。队列是插入和删除操作分别在队列的两端进行的数据结构，它的特点是先进先出。栈和队列在许多应用问题中被不断地用到，对它们的改进和延伸也非常多。

(二) 树结构

树是一种一对多关系的数据结构，表现在父亲节点可以有多个孩子节点，森林是多棵树的组合。树的边数总是比它的顶点数少 1。

树型结构是以分支关系定义的层次结构，它是一种重要的非线性结构。该结构在客观世界中广泛存在，例如人类的家庭族谱、各种社会组织机构、计算机文件管理和信息组织方面都会用到该结构。

树中一个非常重要的特性是树的任意两个节点之间总是恰好存在一条从一个节点到另一个节点的简单路径。树结构多用来描述层次关系，例如，文件目录、组织结构图等。

树的主要应用有状态空间树，在回溯和分支限界章节中将会介绍，这里先不阐述。

树的另一个主要应用是排序树，如二叉查找树、多路查找树等。

(三) 图结构

图是一种比线性表和树更为复杂的数据结构。在线性表中，数据元素之间仅存在线性关系，即每个元素只有一个直接前驱和一个直接后继。在树型结构中，元素之间具有明显的层次关系。每一个元素只能和上一层（如果存在的话）的一个元素相关，但可以和下一层的多个元素相关。在图形结构中，元素之间的关系可以是任意的，一个图中任意两个元素都可以相关，即每个元素可以有多个前驱和多个后继。

图结构描述的是一种多对多的关系，具体表现在图结构包括顶点和边两种元素，刻画图结构需要刻画顶点和顶点、顶点和边之间的关系，所以，一般用邻接链表和邻接矩阵等方法进行刻画。根据图中边的方向性，图可以分为有向图和无向图两种。

如果在图的边上加上权值，这个权值可以表示代价，这时的图就称为加权图，加权图可以用改造后的邻接链表或者邻接矩阵表示。

图的主要特性有连通性和无环性，二者都与路径有关。从图的顶点 u 到顶点 v 的路径可以这样定义：它是图中始于 u 止于 v 的邻接顶点序列。如果是无向图，那么从顶点 u 到顶点 v 的路径和从顶点 v 到顶点 u 的路径是相同的，而有向图却不是这样的。图的连通性是指从某指定顶点到另一指定顶点是否有简单路径，如果有，那么这两点是连通的。连通性在实际应用中有很重大意义。例如，在修建交通设施的时候考虑不同城市之间的连通性，如果我们短期不可能构造全连通的图，可以设置几个重要的枢纽节点，以构造部分连通。

图的无环性与图的回路有关，图的回路是这样一种路径：它的起点和终点是同一个顶点，并且该路径的长度大于 0，同时每边只能出现一次。实际中，我们绕一圈又回到原点构成回路。在不同情况下，图是否包含回路，对所研究的问题将产生非常重要的影响，许多重要的算法要求图是无环图，因为一旦图有回路，算法将不再收敛，而产生无限循环的结果。上节所说的树结构就是一种无环图。

(四) 集合

集合在计算机中一般用序列或者位串表示。序列需要穷举所有的元素，可以采用数组或链表；而位串是用元素个数长的比特串表示元素，如果某元素包含在集合中，则对应的比特位为 1，反之则为 0。

在计算时，对集合的最多操作就是从集合中查找一个元素、增加一个元素或删除一个元素。能够实现这三种操作的数据结构称为字典。如果处理的是动态内容的查找，那么必须考虑字典的查找效率和增、删效率，在实现上需要平衡二者的效率关系。实现字典时，简单的可以用数组实现，如果追求高效，可以使用散列法和平衡查找树等复杂技术实现。

第二节 算法复杂性分析

一、算法的时间复杂性分析与空间复杂性分析

(一) 算法的时间复杂性分析

算法是解决问题的方法。一个问题可以有多种解决方法，不同的算法之间就有了优劣之分。如何对算法进行比较呢？算法可以比较的方面很多，如易读性、健壮性、可维护性、可扩展性等，但这些都不是最关键的方面，算法的核心和灵魂是效率。试想，一个需要运行很多年才能给出正确结果的算法，就算其他方面的性能再好，也是一个不实用的算法。

算法的时间复杂性 (time complexity) 分析是一种事前分析估算的方法。它是对算法所消耗资源的一种渐进分析方法。渐进分析 (asymptotic analysis) 是指忽略具体机器、编程语言和编译器的影响，只关注在输入规模增大时算法运行时间的增长趋势。渐进分析的好处是大大降低了算法分析的难度，是从数量级的角度评价算法的效率的。

1. 输入规模与基本语句

撇开与计算机软硬件有关的因素，影响算法时间代价的最主要的因素是输入规模。输入规模 (input scale) 是指输入量的多少，它可以从问题描述中得到。例如，找出 100 以内的所有素数，输入规模是 100；对一个具有 n 个整数的数组进行排序，输入规模是 n 。一个显而易见的事实是：几乎所有的算法，对于规模更大的输入都需要运行更长的时间。例如，需要更多时间来对更大的数组排序，更大的矩阵转置需要更长的时间。所以运行算法所需要的时间 T 是输入规模 n 的函数，记作 $T(n)$ 。

【例 1-3】 对如下顺序查找算法，请找出输入规模和基本语句。

```
int SeqSearch(int A[], int n, int k)    % 在数组 A[n] 中查找值为 k 的记录
{
    for(int i=0; i<n; i++)
        if(A[i] == k) break;
```