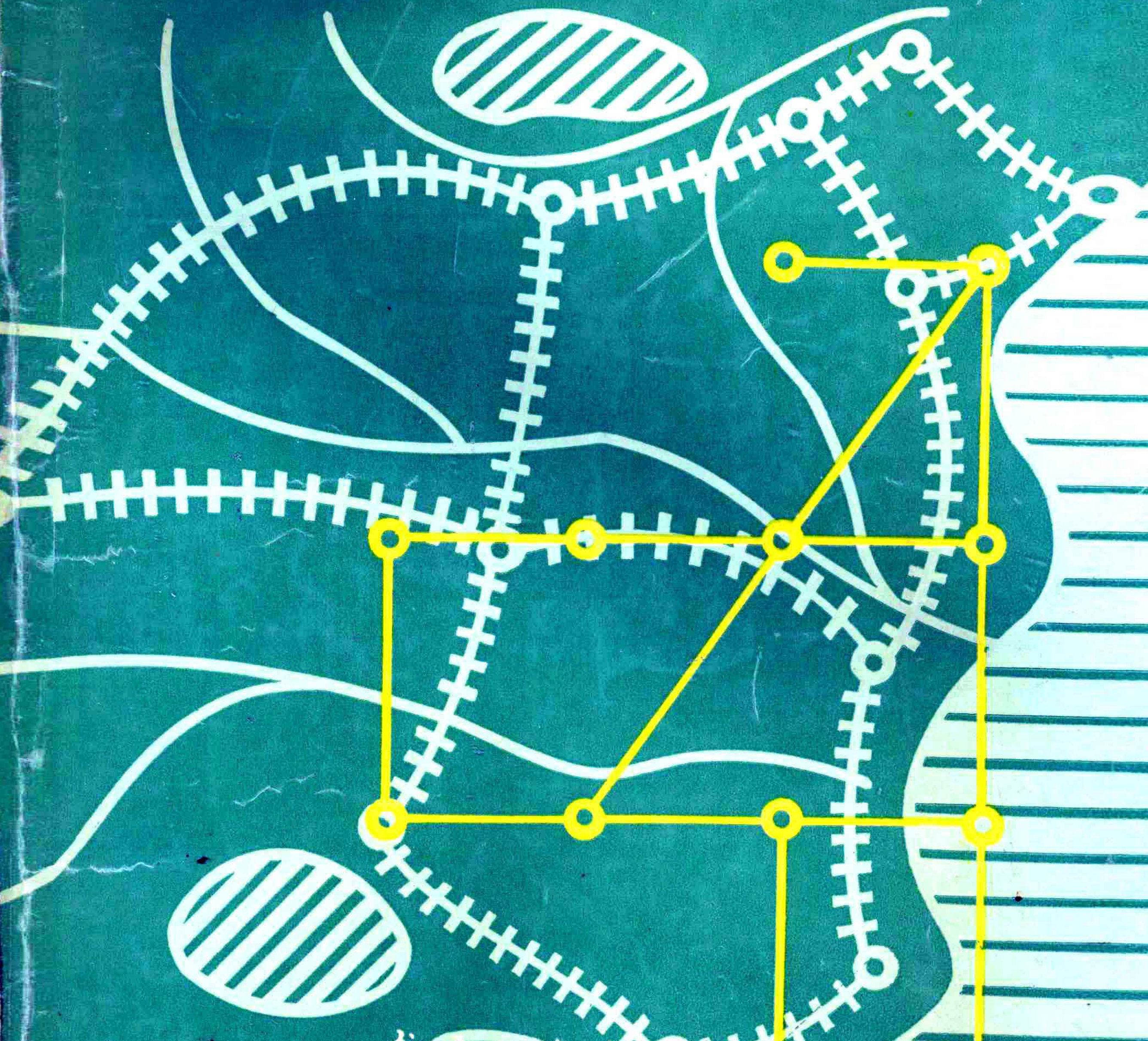
图论及其算法

肖位枢 主编



航空工业出版社

图论及其算法

肖位枢　主编

航空工业出版社

1993

（京）新登字161号

内 容 简 介

本书系统地阐述了图的基本理论并结合实际应用介绍其算法。全书分九章：绪论、基本概念、树与割集、平面图、偶图与匹配问题、图的着色、路径问题、回路问题、网络的流。本着理论与实际密切结合的原则，书中除较详细地进行基本理论的分析外，并用较多篇幅作算法的研究和分析，有较多的例题作为算法的应用实例，每章末还附有较多的习题和思考题作为练习，便于加深理解、巩固提高。

本书是一本理论与应用相结合的基础教材，可作为高等工科院校系统工程、管理工程、自动控制、通讯与计算机科学等专业高年级学生或研究生图论课的教材或教学参考书，也可供有关专业的科技人员自学之用。

图 论 及 其 算 法

肖位枢　主编

航空工业出版社出版发行

（北京市安定门外小关东里14号）

—邮政编码：100029—

全国各地新华书店经售

北京地质印刷厂印刷

1993年7月第1版　　1993年7月第1次印刷

开本：787×1092　1/16　　印张：18.25

印数：1—1800　　字数：453千字

ISBN 7-80046-521-7/TP·036

定价：8.70　元

前　言

图论作为数学的一个分支，已有二百多年的历史，特别是在计算机出现和推动下，图的理论有了迅速的发展，其应用也日愈广泛。现在已成为系统工程、管理工程、计算机科学、通讯与网络理论、自动控制、运筹学以至社会科学等的一种重要的数学工具。图论的应用所以如此广泛，在于它可作为分析处理多种问题的一种较为理想的数学模型，它的算法又可借计算机实现，因而图论与计算机相结合，为图的理论研究和应用开辟了广阔的前景。

本书是编者经过长期的教学实践，在自编教材的基础上，不断加工提炼，反复修改而成。书中除对图的基本理论作了系统地讨论外，并用了较多篇幅介绍图论的各种应用及其算法，举了较多例题作为应用实例及算法说明，目的在于通过对本书的学习，既能掌握图的基本理论，更熟悉一些基本算法，从而可利用计算机解决图论应用的各种实际问题。

鉴于电网络和开关理论内容丰富，已成为图论的一个分支，在工科院校各专业的电工基础或电工学中都有介绍，为了避免重复，不再列入本书内容。因此本书可作为高等工科院校多数专业的通用教材或教学参考书。由于专业要求不同，书中一些理论性较强的内容，如Kuratowsky定理、Vizing定理等的证明可以省略。除理论教学外，如有条件可以结合实际课题安排大作业或上机实习。

本书由肖位枢主编，参加编写的有刘宣（第五章）、肖公信（第四章、第六章）、肖位枢（第一章、第二章、第三章、第七章、第八章、第九章）。

东北大学赵连昌教授在百忙中审阅了本书全稿，并提出了宝贵意见，谨在此表示感谢。

本书在阐述上力求文字简洁、说理清楚、由浅入深、通俗易懂、便于自学，但限于编者水平，书中疏漏或错误之处在所难免，恳请读者多多指正。

肖位枢

1992年12月

目 录

第一章　绪　　论

§1.1　图的图形描述

图的严格定义，我们将在第二章中给出，这里想要指出：社会上和自然界很多问题，如果用图形的方式来描述和分析，不仅形象直观，而且会取得很好的效果，从图形分析上升到理论，就是图论。所以，图论是我们研究和分析问题的一种很好的数学工具或数学模型。为了说明这个问题，下面举几个例子。

例 1.1　有三个酒桶A、B、C，如图1.1，容量分别为8、5、3升，现A桶装满了酒，要求只用这三个桶将酒平分为二，问怎样分法。

如果用语言来描述，一种平分的过程或步骤如下：先将A桶的酒倒入C，再将C桶的酒倒入B，然后依次将A倒入C，C倒入B，B倒入A，C倒入B，A倒入C，最后C倒入B，即可得A、B两桶装的酒各为4升。

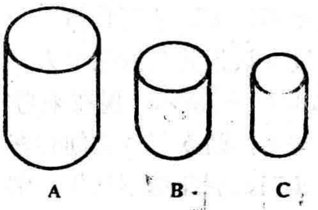


图 1.1

显然，像这样用语言来描述，前后状态和过程不好记，思路容易混淆，如果用图形来描述就会清晰得多。例如：我们用小圆圈（结点）表示各个状态，在小圆圈旁标上（a、b、c）表示这一状态下A、B、C三个桶内分别装有酒a、b、c升，用0——→0表示前一状态经过某一过程（步骤）达到下一状态，并在旁边标上$X \to Y$表示这一过程是将X桶的酒倒入Y桶。

起始状态v_0，标记为（8,0,0），表示A桶装满8升酒，B、C均为空桶，下一步骤只能有两种选择，一种是将A倒入C，即A→C，达到状态v_1（5,0,3），另一种是将A倒入B，即A→B，达到状态u_1（3,5,0），如图1.2所示。假设采用第一种选择，状态v_1的下一步可以是C→B，则达到状态v_2(5,3,0)，如果是A→B，则达到状态q(0,5,3)，但是，状态q的下一步只能回到状态v_1或者到状态u_1，这样的过程是不合理的，这就说明状态v_1的下一步只能是经C→B而达到状态v_2，同理v_2的下一步只能是经A→C而达到状态v_3，

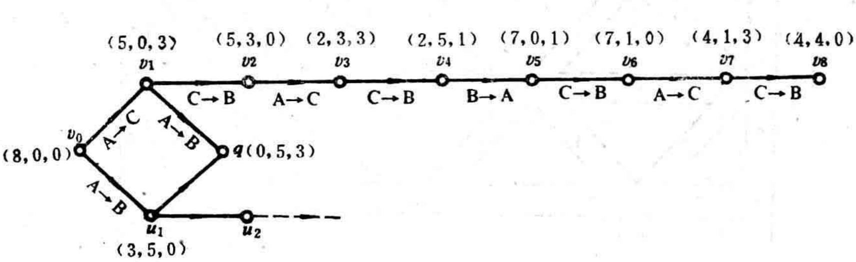


图 1.2

依此类推最后达到状态v_8（4,4,0），过程即告结束，这时A、B桶内的酒各为4升。

如果采用第二种选择，即由初始状态经A→B达到状态u_1（3,5,0），最后也可实现将酒平分为二，而且所需步骤少于第一种选择，它的图形描述，留给读者。

由此可见，这一问题用图形来描述，不仅形象直观，而且每一步都非常严格，符合逻辑思维和分析的规律，问题怎样解，有多少种解法，哪一种解法最佳，都清楚地呈现出来，使我们对这一问题的解，获得一清晰完整的概念。

例 1.2 试证：任意六个人在一起，其中一定有三个人彼此互相认识，或者有三个人彼此都不认识。

可以用图形描述法证明这个命题。用6个小圆点a，b，c，d，e，f表示任意六个人，如果某两人彼此认识，则在相应的两点之间联一实线，如果两人彼此不认识，则在相应两点之间画一条虚线，因此，任意两点之间如果不存在实线，就一定存在虚线，反之亦然。于是，这一命题的图形表示就是：一定存在一个实线三角形，或者一定存在一个虚线三角形。

六人中的任意一个人，比如a，他对其余五个人来说，最少认识其中三个，否则最少有三个不认识，这两种情况一定有一种而且只能有一种存在。

设a认识其余的三个人，比如b、c、d，则a点与这三个点之间，可联一条实线如图1.3（a）所示。现在来考察b、c、d，如果他们三个人中至少有两人彼此认识，比如b与c，则b与c之间存在一条实线，于是a、b、c三点构成一实线三角形。若b、c、d三人彼此都不认识，则b、c、d构成一虚线三角形，因此，b、c、d之间的任何一种情况都将导致出现实线三角形或者虚线三角形。

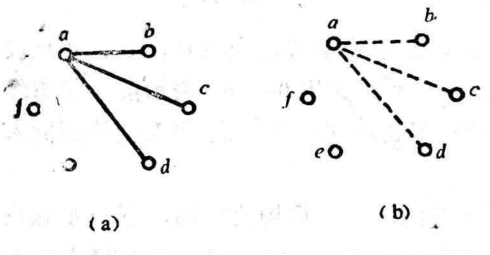


图 1.3

反之，若a对b、c、d都不认识，则a与这三点之间可联一条虚线，如图1.3(b)所示，可用同样方法证明必然存在实线三角形或者虚线三角形。因此命题得证。

例 1.3 图1.4表示某一城市部分街道，图中标的数字表示街道的长度。问从学校A到医院H，怎样走路程最短。

这是一个求最短路径的问题，答案是沿着路径：A（学校）→D→F→G→J→H(医院)，则所走的路程最短。这一答案是怎样得

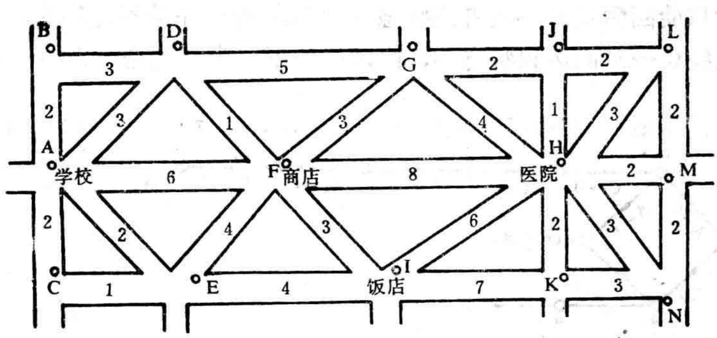


图 1.4

到的，我们将在第七章中详细分析并给出有效算法。

上面列举的例子，只是从几个具体问题说明如何用图形描述和分析，当然不可能勾勒出图论研究问题的全貌，但从中我们可以看到，这些问题都可以看作某些事物以及它们之间存在着的某种联系，图论就是研究事物及它们之间联系的一门学科，正如用函数的图象描述函数一样，在这里，我们用图形来描述所研究的事物及它们之间的联系，即用图形来描述图。由于用图形描述图，形象直观，便于理解，因而是图的一种较好的表达形式，所以图论的很多名词和术语，都采用了图形的名词和术语，例如把事物称之为点，而用边表示事物之间的联系，等等。

图论研究的事物可以是具体的客观实体，也可以是抽象的概念，如同集合中的元素一样，事物之间的联系，可以是静态的关系，也可以是动态的关系，所以，图论中的点和边，不一定需要用平面上的几何点和线来表示，就是说不一定需要用图形来描述图，正如不一定需要用图象描述函数一样。对于图的定义可以进一步抽象化，对它进行高度概括，使它更具有普遍意义，这就是我们以后要加以阐述的。但是，为了对问题的分析更为直观清晰，我们以后在研究图时，仍经常伴以图形描述。

由于图是研究事物及它们之间关系的学科，因此任何一个能用二元关系描述的系统，都可以用图提供数学模型，因此图论模型具有广泛的适用性，特别是计算机出现以后，图论的很多复杂问题，借助计算机得到了圆满的解决，使图论的理论研究和应用，获得了更为迅速的发展。

§1.2 图的拓扑变换

图 1.5 是某地区铁路线路图。图 1.6 是对这个线路图进行简化，得出的一个示意图。

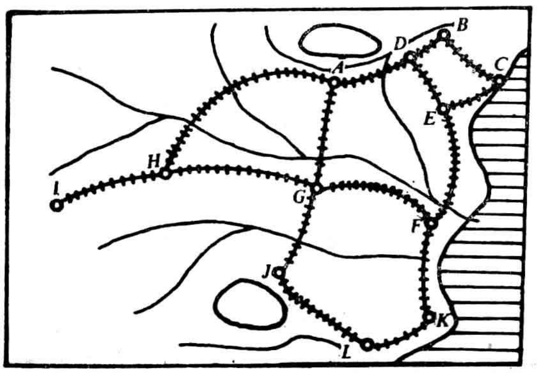


图 1.5

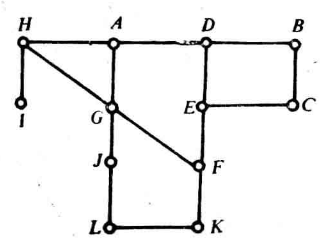


图 1.6

可以看出，示意图与原来的线路图，在外观上有了很大的变动，原来铁路线的长度、方向以及各站的位置，在示意图中都变更了，唯一保持不变的，就是各站之间的联接关系，这一关系表示各站之间的连通性，就是站与站之间，是否有线路连通，并且由某一站至另一站，是否须经别的站。

通常，旅客乘火车时所关心的事情，并不是两站之间的距离和方向等，而是关心在哪一站上车，经过哪些站便可抵达目的地。如果是这样，那么，用示意图代替原来的线路

图，便可达到这一目的，使图形更为清晰简单。

由图 1.5 的实际线路图变成图1.6的示意图，是经过一种几何变换的，这种几何变换的主要特点，就是保持物体原来的连通性，就是说，物体本来是连着的部分，不会因变换而断开，本来不是连着的部分，也不会因变换而接合起来。一个具有保持连通性的变换，称为拓扑变换。两个图形，能够通过拓扑变换变成同一形状的图形，称为是拓扑等价的。

图1.7（a）中的图形是拓扑等价的，同理（b）中的图形也是拓扑等价的，而（c）

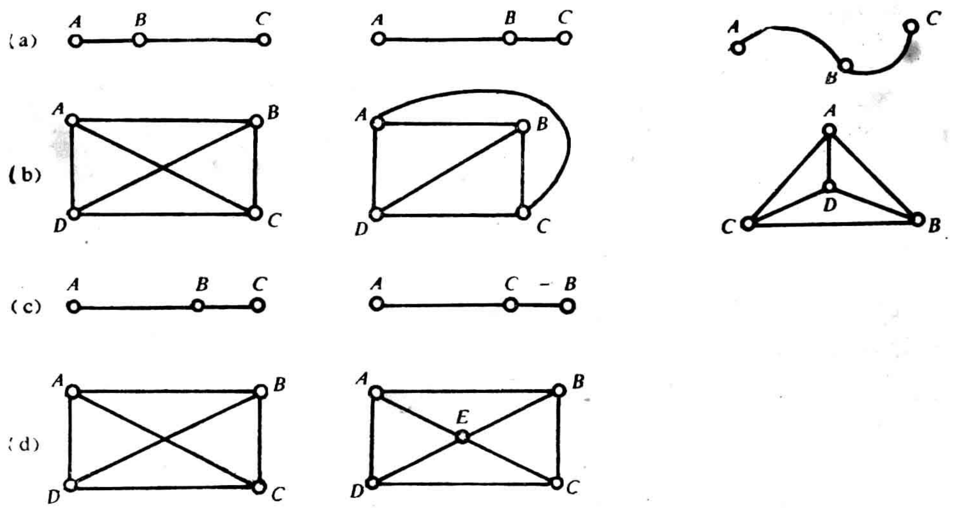


图 1.7

（d）中的图形则不是拓扑等价的。

图的拓扑变换，可以设想为图是由橡皮筋构成的，我们可以任意把它拉长、扭曲，但是却不能把它扯断或接合起来。

用以表示图的图形，也具有这种拓扑变换的特性，所以点和边的位置、形状是无关紧要的，我们关心的只是点与点之间的联接关系，因此，同样性质的图可以画成多种不同的形式，只须保持点和边的关系不变即可。

§1.3 图的计算复杂性

图作为一种数学模型，必然涉及对它的计算，同一个问题，可能有不同的算法，算法不同，效率也会不一样，因此，对一个算法，人们常用计算复杂性去衡量它的效率或计算的难度。评价一个算法计算复杂性的标准，是这个算法需要耗费的时间和空间。如果求解同一问题，算法 A_1比算法 A_2需要的时间和空间少，就说算法 A_1比 A_2好，或 A_1比 A_2的效率高。不言而喻，当求解一个问题时，我们总是希望找到一个效率高的算法。

通常，对一个算法，人们更关心的是时间的耗费，即计算的时间复杂性，而对空间复杂性较少讨论。一个算法的时间复杂性可以简单地表为从输入数据到计算出结果所需的时间或计算的步数（即需要计算少多步才可得出结果）。它是输入数据（初始数据）量的函数。输入数据量常用一个正整数来表示，也称为问题的规模（或大小）。例如，在图论中，

问题规模决定于图的边数或结点数。在计算矩阵乘法时，问题的规模可以用矩阵的阶数来定义。

设输入数据量为 n，则算法 A 的时间复杂性用 $T_A(n)$ 表示，在不致引起混淆时，可将下标 A 去掉。当输入量 n 逐渐增大时，时间复杂性的极限称为算法的渐近时间复杂性，它决定了能够处理的问题的最大规模。

在复杂性理论中，函数的量级具有很重要的意义，下面我们给出它的定义。

定义 1.1 给定两个自然数 n 的函数 $F(n)$ 与 $G(n)$，当且仅当存在一个正常数 K 和一个 n_0，使得 $n \geqslant n_0$ 时都有 $F(n) \leqslant KG(n)$，则称函数 $F(n)$ 以函数 $G(n)$ 为界，记作 $F(n)=O(G(n))$，或称 $F(n)$ 是 $O(G(n))$。

这里"O"表示数量级的概念。例如对某个常数 $K>0$，一个算法能在 Kn^2 的时间内处理完规模为 n 的输入，就说这个算法的时间复杂性是 $O(n^2)$，读作"n 平方级"的。

例 1.4 对于下面三个简单的程序段：

（a）
```
X:=X+1
```
（b）
```
FOR i:=1 TO n DO
    X:=X+1;
```
（c）
```
FOR i:=1 TO n DO
FOR j:=1 TO n DO
    X:=X+1;
```

程序（a）只运算一步，执行时间是个常量；程序（b）的语句要执行 n 次，计算的时间与 n 成正比，而程序（c）的计算时间与 n^2 成正比，因此，三个程序段计算时间复杂性分别是 $O(1)$、$O(n)$ 和 $O(n^2)$。

在研究算法复杂性时，算法的渐近时间复杂性是判定算法好坏的一个重要标准，这时，在确定复杂性的量级时，函数的低次项可以忽略，于是多项式（$3n^3+6n^2+n+6$）是 $O(n^3)$ 而 $\frac{1}{2}n^2$ 是 $O(n^2)$，显然，这样表示是很简便的。

当需要比较两个复杂性函数的量级时，常采用下面定义的方法：

定义 1.2 设 $F(n)$ 与 $G(n)$ 为自然数 n 的两个函数，令 $\lim\limits_{n\to\infty} F(n)/G(n)=L$，如果

（i）$L=a$，a 为有限正常量，则称 $F(n)$ 与 $G(n)$ 同量级。

（ii）$L=0$，则称 $F(n)$ 的量级比 $G(n)$ 的量级低。

（iii）$L\to\infty$，则称 $G(n)$ 的量级比 $F(n)$ 的量级低。

例 1.5 对以下四种情况，比较两个函数的量级。

（a）设 $F(n)=3n^2-4n+2, G(n)=\frac{1}{2}n^2$.

则 $L=6$，因此两个函数同量级。

（b）设 $F(n)=\log_2 n$，$G(n)=n$，

则 $L=\lim\limits_{n\to\infty}\frac{ln\ n}{n}\cdot\log_2 e=\lim\limits_{n\to\infty}\frac{\log_2 e}{n}=0$

这里我们用了如下法则，即如果

$$\lim_{n\to\infty} F(n)=\lim_{n\to\infty} G(n)=\infty$$

假定有导数 $F'(n)$ 及 $G'(n)$，且极限存在，则

$$\lim_{n\to\infty} F(n)/G(n) = \lim_{n\leftarrow\infty} F'(n)/G'(n)$$

因 $L=0$，我们即得 $\log_2 n$ 的量级低于 n 。

（c）设 $F(n)=a^n$，$G(n)=n^k$，这里 a 、k 为任意给定的常量，且都大于 1 。

令 $U(n)=F(n)/G(n)$，则

$$U(n+1)/U(n)=a(n/(n+1))^k$$

当固定 k 时，我们总可以找到 n 的一个足够大的值，比如说 n_0，使 $n>n_0$ 时，$(n/(n+1))^k \simeq 1$，即

$$U(n+1)\simeq a\cdot U(n)$$

于是当 $n\geqslant n_0$ 时

$$\begin{aligned} U(n) &= a^n/n^k \\ &\simeq a^n/n_0^k = a^{n-n_0}\cdot U(n_0) \end{aligned}$$

则 $L=\lim\limits_{n\to\infty} U(n)=\infty$

由此即得：阶为 n 的指数函数的量级将高于 n 的任何多项式的量级。所以算法为多项式阶的算法是我们所希望得到的，而指数阶的算法，则应尽量避免。

（d）如果 $F(n)$ 和 $G(n)$ 如（c）所设，且 $H(n)=n!$，用类似（c）的同样近似法，读者可以验证 $H(n)$ 的量级比 $F(n)$ 和 $G(n)$ 的量级都高。即 n 的阶乘，它的量级高于 n 的多项式，也高于 n 次指数函数的量级。

表1.1列出了六种算法的时间复杂性函数在不同问题规模 n 时，需要的计算步数。

表 1.1

问题规模 n / 时间复杂性	2	8	128	1024
n	2	2^3	2^7	2^{10}
$n\log_2 n$	2	3×2^3	7×2^7	10×2^{10}
n^2	2^2	2^6	2^{14}	2^{20}
n^3	2^3	2^9	2^{21}	2^{30}
2^n	2^2	2^8	2^{128}	2^{1024}
n!	2	5×2^{13}	5×2^{714}	7×2^{8766}

计算时间是与计算步数成比例的，设计算机的计算速度为 2^{20} 步/秒，时间换算如下：

$$\begin{aligned} 2^{20}\text{步/秒} &\simeq 0.9\times2^{26}\text{步/分} \\ &\simeq 0.9\times2^{32}\text{步/小时} \\ &\simeq 1.3\times2^{36}\text{步/日} \\ &\simeq 0.9\times2^{45}\text{步/年} \\ &\simeq 0.7\times2^{52}\text{步/世纪} \end{aligned}$$

设问题规模 $n=128$，则时间复杂性函数为 n^2 的算法，需要的计算时间只有 1/64 秒，n^3 需要的计算时间也不过 2 秒，而 2^n 需要的计算时间超过 2^{76} 世纪，$n!$ 需要的计算时间超过 5×2^{662} 世纪，显然，后两种算法是不可能在实际有限的时间内完成的。

由此可见，一个算法的时间复杂性函数的量级具有多么重要的意义，它是反映算法性能的重要指标。对于某一个问题而言，如果算法的时间复杂性函数的量级越低，说明算法的效率越高。

可能认为，现代计算机发展突飞猛进，计算速度成百成千倍增加，算法效率的高低已没有多大意义，其实不然，表1.2给出了提高计算机运算速度对列出的算法带来处理能力增加的情况。

表 1.2

算法	时间复杂性	提高速度前单位时间能处理数据量	提高速度 2^3 倍后单位时间能处理数据量	提高速度 2^7 倍后单位时间能处理数据量	提高速度2^{10} 倍后单位时间能处理数据量
A_1	n	N_1	$8N_1$	$128N_1$	$1024N_1$
A_2	n^2	N_2	$2.8N_2$	$11.3N_2$	$32N_2$
A_3	2^n	N_3	N_3+3	N_3+7	N_3+10
A_4	8^n	N_4	N_4+1	$N_4+2.3$	$N_4+3.3$

从表1.2可以看出，算法 A_1 在同一时间里能处理的输入量增加的倍数与计算机速度提高的倍数相同，算法A_2的处理能力，也比原来增加若干倍，而算法 A_3 和 A_4 就很差，处理能力仅增加若干个，例如算法 A_4，当计算机运算速度提高 1024 倍时，它能处理的数据量，比原先只多不到 4 个。这就说明，对于多项式函数的算法，提高计算机运算速度，对提高计算效率，还能收到较为明显的效果，而对于指数函数的算法，就没有多大意义。因此一般认为，如果一个算法的时间复杂性是以多项式为界的，则是一个有效的算法，对于一个有效的算法，那怕输入的数据量很大，也认为计算机可以处理得了，而对于指数函数的算法，则认为是一个低效率的算法，只要问题的输入量稍微大一些，计算机就无法处理了。

以上是从计算复杂性量级的高低衡量算法的优劣，但是也要指出，当问题的输入量较小时，或者带有不同大小的常量因子时，计算复杂性量级高的算法，可能反而比计算复杂性量级低的算法要好，试看下面的例子。

例 1.6 设算法 A_1、A_2、A_3、A_4、A_5的时间复杂性为：

$T_1(n)=1000\,n$

$T_2(n)=100\,n\log_2 n$

$T_3(n)=10\,n^2$

$T_4(n)=n^3$

$T_5(n)=2^n$

则当：$2\leqslant n\leqslant 9$ 时，A_5比 A_1、A_2、A_3、A_4都好。

$10\leqslant n\leqslant 58$ 时，A_3比 A_1、A_2、A_4、A_5都好。

$59\leqslant n\leqslant 1024$ 时，A_2比其它算法好。

$n>1024$ 时，算法 A_1最好。

所以，研究算法复杂性时，不仅要看复杂性函数的量级，也应注意所含的常数因子以及实际问题的规模。

习题与思考题

1．某人挑一担青菜、牵一条狗、一只羊赶路。途经一条小河，船小一次只容人带狗、

羊、菜三者之一过河，当人不在场时，应避免狗和羊或羊和菜在一起，求过河的方案。

2．选手A、B轮流从10根火柴中抽取火柴，每次只准取1或2根，不准不取也不准多取，最后取完火柴者获胜。问怎样取方可取胜。

3．有4件产品，其中有一件不合格，但它的外形并无差异，只是重量不合标准。问如何以最少的次数用天秤找出这件不合格的产品，并判别出它的重量比标准产品重还是轻。

4．甲、乙两人进行网球比赛，如果某人连胜两局或总计胜三局，他就获胜，比赛即告结束，试分析各种取胜的比赛情况。

5．试判别图1.8中的图形，哪些是彼此拓扑等价的。

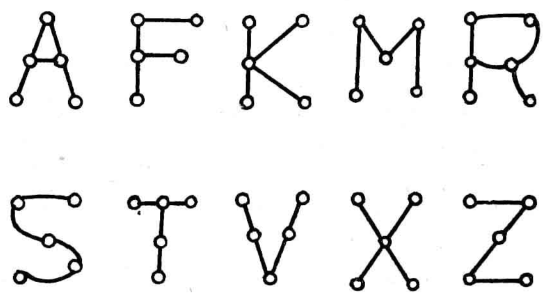


图 1.8

6．图1.9中的任意两个图形，是否都是拓扑等价的。

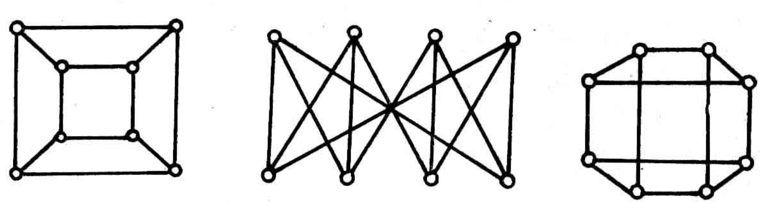


图 1.9

7．如果对于某一$k>0$及存在$n_0>0$，当$n>n_0$时，有$F(n)\geqslant k$且存在常数$C_1>0$和$C_2>0$，使得当$n>n_0$时，有

$$G(n)\leqslant C_1F(n)+C_2$$

证明$G(n)$是$O(F(n))$。

8．对下面的函数对，确定最小的整数值$n_0>0$，使得当$n\geqslant n_0$时每对中的第一个函数恒大于等于第二个函数。

（1）n^2，$10n$。

（2）2^n，$2n^3$。

（3）$n^2/\log_2 n$，$n(\log_2 n)^2$。

（4）$n^3/2$，$n^{2.81}$。

9．如果存在一个正的常数c，使得一切$n\geqslant 0$，有$F(n)\leqslant CG(n)$，则记为$F(n)\leqslant G(n)$。试证明：若$F_1\leqslant G_1$和$F_2\leqslant G_2$，必有

$$F_1+F_2\leqslant G_1+G_2$$

关系“$\leqslant$”还有一些什么性质？

第二章　图的基本概念

§2.1　图 与 子 图

一、有向图与无向图

第一章讲了如何用图形描述图，同时也指出图如同函数一样，也是一个数学抽象。这里我们从集合论的角度来给图下定义。

定义 2.1　一个有向图D是一个三元组（V,E,f），其中V是一个非空集合，它的元素称为有向图D的结点，E是一个集合，它的元素称为有向图D的弧（边），f是从E到$V\times V$上的一个映射（函数）。

例 2.1　设$V=\{a,b,c,d\}$，$E=\{e_1,e_2,e_3,e_4,e_5,e_6\}$，且$f(e_1)=\langle a,b\rangle$，$f(e_2)=\langle c,b\rangle$，

$$f(e_3)=\langle b,c\rangle,\ f(e_4)=\langle c,d\rangle,$$
$$f(e_5)=\langle d,b\rangle,\ f(e_6)=\langle d,d\rangle。$$

则$D=(V,E,f)$是一个有向图，它的图形如图 2.1 所示。

例 2.2　设有向图$D=(V,E,f)$的图形如图 2.2 所示，则可得：

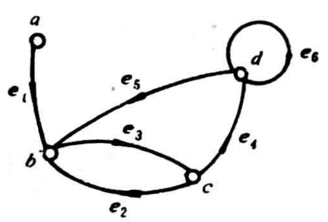


图 2.1

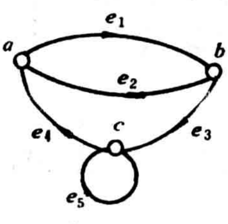


图 2.2

$$V=\{a,\ b,\ c\}$$
$$E=\{e_1,\ e_2,\ e_3,\ e_4,\ e_5\}$$
$$f(e_1)=\langle a,\ b\rangle,\ f(e_2)=\langle b,\ a\rangle$$
$$f(e_3)=\langle b,\ c\rangle,\ f(e_4)=\langle c,\ a\rangle,\ f(e_5)=\langle c,\ c\rangle.$$

从定义可知，E中的元素总是与V中的序偶有着对应的关系，因此可用V中的序偶代替E中的元素。如例2.2，可写成$E=\{\langle a,\ b\rangle,\ \langle b,\ a\rangle,\ \langle b,\ c\rangle,\ \langle c,\ a\rangle,\ \langle c,\ c\rangle\}$。一个有向图$D$，可简记为（$V,\ E$）。仿此可以对无向图定义如下：

定义 2.2　一个无向图G记作$G=(V,\ E)$，其中V是一个非空集合，它的元素称为图的结点，E是V中的无序偶集合，它的元素称为图的边。

例 2.3　设$V=\{a,\ b,\ c,\ d\}$，$E=\{(a,\ b),\ (b,\ b),\ (b,\ c),\ (c,\ d),(d,\ a),\ (a,\ d)\}$，则

$G=(V, E)$是一个无向图，它的图形如图 2.3所示。

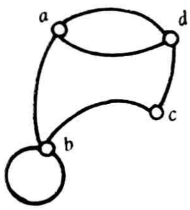


图 2.3

无向图与有向图的区别，在于E中的元素是无序偶还是有序偶。从图形上看，无向图的边是无向边，有向图的边是有向边。一个有向图$D=(V,E)$，如果将E中的有序偶元素改为无序偶元素，变成无向图G，称为有向图D的基础图（或底图）。从图形上看，将有向图边的箭头去掉，得到的无向图就是它的基础图。有向图和无向图，统称为图。

在图中，边e的两个端点a，b称为e的关联点，并称点a和b是邻接的，不与任何结点邻接的结点称为孤点，只有孤点的图称为零图，只有一个孤点的图称为平凡图。结点集合V是有限集合的图称为有限图，否则称为无限图，今后我们只讨论有限图，并简称为图。从结点到自身的边称为自环；在有向图中两结点之间同一方向的边称为平行边，对于无向图，关联于相同两结点之间的边称为平行边；含平行边的图称为多重边图，不含平行边和自环的图称为简单图；结点的数目称为图的阶。在无向图中，与结点v关联的边的数目称为结点v的次数，记作$deg(v)$；在有向图中，从结点v引出的弧的数目称为v的引出次数，记作$deg^+(v)$，引向v的弧的数目称为v的引入次数，记作$deg^-(v)$，v的引出次数与引入次数的和称为结点v的次数，记作$deg(v)$。

例 2.4 如图2.3的无向图，$deg(b)=4, deg(a)=3$。如图2.1的有向图，$deg^+(b)=1$，$deg^-(b)=3$，故$deg(b)=deg^+(b)+deg^-(b)=1+3=4$，而$deg^+(d)=2$，$deg^-(d)=2$，故$deg(d)=4$。

一个图常用(n, m)表示，其中n为图的结点数，m为图的边数，即$n=|V|, m=|E|$。

无论是有向图还是无向图，下面两条定理，都表明了结点次数的特性。

定理 2.1 任意一个图(n, m)，结点次数的总和等于边数的二倍，即

$$\sum_{i=1}^{n} deg(v_i)=2m \qquad (2.1)$$

证：因一条边与两个结点关联，出现一条边就使结点的总次数增加2，因此边数的二倍就是结点的总次数。 ∎

定理 2.2 任意一个图，次数为奇数的结点数必为偶数。

证：设V_1和V_2分别为图中次数为奇数和偶数的结点集合，因而有$V_1\cup V_2=V$，$V_1\cap V_2=\phi$。则

$$\sum_{v\in V} deg(v)=\sum_{v\in V_1} deg(v)+\sum_{v\in V_2} deg(v)=2m$$

因$2m$是偶数，$\sum_{v\in V_2} deg(v)$是偶数之和必为偶数，故$\sum_{v\in V_1} deg(v)$也应是偶数，但V_1中每个结点次数都是奇数，因此必须有偶数个奇次结点，才可能使它们的次数和为偶数，命题得证。

二、正则图与完全图

定义 2.3 一个简单无向图$G=(V, E)$。

1）如果每个结点都有相同次数d，称G为d次正则图。