21st CENTURY
实用规划教材

21世纪全国应用型本科计算机案例型规划教材

赠送 电子课件 www.pup6.com

赠送 教学参考资料 www.pup6.com

图论算法理论、实现及应用

Graph Theory

主 编 王桂平 王 衍 任嘉辰

[redacted]

北京大学出版社
PEKING UNIVERSITY PRESS

21 世纪全国应用型本科计算机案例型规划教材

图论算法理论、实现及应用

王桂平　王　衍　任嘉辰　主　编

北京大学出版社
PEKING UNIVERSITY PRESS

内容简介

本书系统地介绍了图论算法理论，并选取经典的ACM/ICPC竞赛题目为例题阐述图论算法思想，侧重于图论算法的程序实现及应用。本书第1章介绍图的基本概念和图的两种存储表示方法：邻接矩阵和邻接表，第2～9章分别讨论图的遍历与活动网络问题，树与图的生成树，最短路径问题，可行遍性问题，网络流问题，支配集、覆盖集、独立集与匹配，图的连通性问题，平面图及图的着色问题等。

本书可以作为高等院校计算机(或相关专业)图论等相关课程的主教材，也可作为ACM/ICPC竞赛的辅导教材。

图书在版编目(CIP)数据

图论算法理论、实现及应用/王桂平，王衍，任嘉辰主编. —北京：北京大学出版社，2011.1

(21世纪全国应用型本科计算机案例型规划教材)

ISBN 978-7-301-17578-1

Ⅰ. ①图…　Ⅱ. ①王…②王…③任…　Ⅲ. ①图论算法—算法程序—高等学校—教材　Ⅳ. ①0157.5

中国版本图书馆CIP数据核字(2010)第146867号

书　　名：图论算法理论、实现及应用
著作责任者：王桂平　王　衍　任嘉辰 主编
责 任 编 辑：郑　双
标 准 书 号：ISBN 978-7-301-17578-1/TP・1122
出　版　者：北京大学出版社
地　　址：北京市海淀区成府路205号　邮编：100871
网　　址：http://www.pup.cn　http://www.pup6.com
电　　话：邮购部62752015　发行部62750672　编辑部62750667　出版部62754962
电 子 邮 箱：pup_6@163.com
印　刷　者：河北滦县鑫华书刊印刷厂
发　行　者：北京大学出版社
经　销　者：新华书店
787毫米×1092毫米　16开本　30.5印张　705千字
2011年1月第1版　　2011年1月第1次印刷
定　　价：54.00元

信息技术的案例型教材建设

(代丛书序)

刘瑞挺

北京大学出版社第六事业部在 2005 年组织编写了《21 世纪全国应用型本科计算机系列实用规划教材》，至今已出版了 50 多种。这些教材出版后，在全国高校引起热烈反响，可谓初战告捷。这使北京大学出版社的计算机教材市场规模迅速扩大，编辑队伍茁壮成长，经济效益明显增强，与各类高校师生的关系更加密切。

2008 年 1 月北京大学出版社第六事业部在北京召开了“21 世纪全国应用型本科计算机案例型教材建设和教学研讨会”。这次会议为编写案例型教材做了深入的探讨和具体的部署，制定了详细的编写目的、丛书特色、内容要求和风格规范。在内容上强调面向应用、能力驱动、精选案例、严把质量；在风格上力求文字精练、脉络清晰、图表明快、版式新颖。这次会议吹响了提高教材质量第二战役的进军号。

案例型教材真能提高教学的质量吗？

是的。著名法国哲学家、数学家勒内・笛卡儿(Rene Descartes，1596—1650)说得好：“由一个例子的考察，我们可以抽出一条规律。(From the consideration of an example we can form a rule.)”事实上，他发明的直角坐标系，正是通过生活实例而得到的灵感。据说是在 1619 年夏天，笛卡儿因病住进医院。中午他躺在病床上，苦苦思索一个数学问题时，忽然看到天花板上有一只苍蝇飞来飞去。当时天花板是用木条做成正方形的格子。笛卡儿发现，要说出这只苍蝇在天花板上的位置，只需说出苍蝇在天花板上的第几行和第几列。当苍蝇落在第四行、第五列的那个正方形时，可以用(4，5)来表示这个位置……由此他联想到可用类似的办法来描述一个点在平面上的位置。他高兴地跳下床，喊着“我找到了，找到了”，然而不小心把国际象棋撒了一地。当他的目光落到棋盘上时，又兴奋地一拍大腿：“对，对，就是这个图”。笛卡儿锲而不舍的毅力，苦思冥想的钻研，使他开创了解析几何的新纪元。千百年来，代数与几何，井水不犯河水。17 世纪后，数学突飞猛进的发展，在很大程度上归功于笛卡儿坐标系和解析几何学的创立。

这个故事，听起来与阿基米德在浴池洗澡而发现浮力原理，牛顿在苹果树下遇到苹果落到头上而发现万有引力定律，确有异曲同工之妙。这就证明，一个好的例子往往能激发灵感，由特殊到一般，联想出普遍的规律，即所谓的“一叶知秋”、“见微知著”的意思。

回顾计算机发明的历史，每一台机器、每一颗芯片、每一种操作系统、每一类编程语言、每一个算法、每一套软件、每一款外部设备，无不像闪光的珍珠串在一起。每个案例都闪烁着智慧的火花，是创新思想不竭的源泉。在计算机科学技术领域，这样的案例就像大海岸边的贝壳，俯拾皆是。

事实上，案例研究(Case Study)是现代科学广泛使用的一种方法。Case 包含的意义很广：包括 Example 例子，Instance 事例、示例，Actual State 实际状况，Circumstance 情况、事件、境遇，甚至 Project 项目、工程等。

我们知道在计算机的科学术语中，很多是直接来自日常生活的。例如 Computer 一词早在 1646 年就出现于古代英文字典中，但当时它的意义不是“计算机”而是“计算工人”，

即专门从事简单计算的工人。同理，Printer 当时也是“印刷工人”而不是“打印机”。正是由于这些“计算工人”和“印刷工人”常出现计算错误和印刷错误，才激发查尔斯·巴贝奇(Charles Babbage，1791—1871)设计了差分机和分析机，这是最早的专用计算机和通用计算机。这位英国剑桥大学数学教授、机械设计专家、经济学家和哲学家是国际公认的“计算机之父”。

20 世纪 40 年代，人们还用 Calculator 表示计算机器。到电子计算机出现后，才用 Computer 表示计算机。此外，硬件(Hardware)和软件(Software)来自销售人员。总线(Bus)就是公共汽车或大巴，故障和排除故障源自格瑞斯·霍普(Grace Hopper，1906—1992)发现的“飞蛾子”(Bug)和“抓蛾子”或“抓虫子”(Debug)。其他如鼠标、菜单……不胜枚举。至于哲学家进餐问题，理发师睡觉问题更是操作系统文化中脍炙人口的经典。

以计算机为核心的信息技术，从一开始就与应用紧密结合。例如，ENIAC 用于弹道曲线的计算，ARPANET 用于资源共享以及核战争时的可靠通信。即使是非常抽象的图灵机模型，也受到二战时图灵博士破译纳粹密码工作的影响。

在信息技术中，既有许多成功的案例，也有不少失败的案例；既有先成功而后失败的案例，也有先失败而后成功的案例。好好研究它们的成功经验和失败教训，对于编写案例型教材有重要的意义。

我国正在实现中华民族的伟大复兴，教育是民族振兴的基石。改革开放 30 年来，我国高等教育在数量上、规模上已有相当的发展。当前的重要任务是提高培养人才的质量，必须从学科知识的灌输转变为素质与能力的培养。应当指出，大学课堂在高新技术的武装下，利用 PPT 进行的“高速灌输”、“翻页宣科”有愈演愈烈的趋势，我们不能容忍用“技术”绑架教学，而是让教学工作乘信息技术的东风自由地飞翔。

本系列教材的编写，以学生就业所需的专业知识和操作技能为着眼点，在适度的基础知识与理论体系覆盖下，突出应用型、技能型教学的实用性和可操作性，强化案例教学。本套教材将会有机融入大量最新的示例、实例以及操作性较强的案例，力求提高教材的趣味性和实用性，打破传统教材自身知识框架的封闭性，强化实际操作的训练，使本系列教材做到“教师易教，学生乐学，技能实用”。有了广阔的应用背景，再造计算机案例型教材就有了基础。

我相信北京大学出版社在全国各地高校教师的积极支持下，精心设计，严格把关，一定能够建设出一批符合计算机应用型人才培养模式的、以案例型为创新点和兴奋点的精品教材，并且通过一体化设计、实现多种媒体有机结合的立体化教材，为各门计算机课程配齐电子教案、学习指导、习题解答、课程设计等辅导资料。让我们用锲而不舍的毅力，勤奋好学的钻研，向着共同的目标努力吧！

刘瑞挺教授　本系列教材编写指导委员会主任、全国高等院校计算机基础教育研究会副会长、中国计算机学会普及工作委员会顾问、教育部考试中心全国计算机应用技术证书考试委员会副主任、全国计算机等级考试顾问。曾任教育部理科计算机科学教学指导委员会委员、中国计算机学会教育培训委员会副主任。PC Magazine《个人电脑》总编辑、CHIP《新电脑》总顾问、清华大学《计算机教育》总策划。

前　　言

一、图论研究及图论教学①

图论(Graph Theory)是数学的一个分支，它以图为研究对象。图论中的图是由若干个给定的顶点及若干条连接两个顶点的边所构成的图形，这种图形通常用来描述某些事物之间的某种特定关系，用顶点代表事物，用连接两个顶点的边表示相应两个事物间具有这种关系。这种图提供了一个很自然的数据结构，可以对自然科学和社会科学中许多领域的问题进行恰当的描述或建模，因此图论研究越来越得到这些领域的专家和学者的重视。

图论最早的研究源于瑞士数学家莱昂哈德•欧拉(Leonhard Euler, 1707—1783 年)，他在 1736 年成功地解决了**哥尼斯堡(Königsberg)七桥问题**，从而开创了图论的研究。

哥尼斯堡七桥问题。东普鲁士哥尼斯堡市(今俄罗斯加里宁格勒)有一条布格(Pregel)河，如图 1(a)所示。布格河横贯哥尼斯堡城区，它有两条支流，在这两条支流之间夹着一块岛形地带，这里是城市的繁华地区。全城分为北、东、南、岛 4 个区，各区之间共有 7 座桥梁相联系。

人们长期生活在河畔、岛上，来往于七桥之间。有人提出这样一个问题：能不能一次走遍所有的 7 座桥，而每座桥只准经过一次？问题提出后，很多人对此很感兴趣，纷纷进行试验，但在相当长的时间里，始终未能解决。

欧拉在 1736 年解决了这个问题，他将这个问题抽象为一个图论问题：把每一块陆地用一个顶点来代替，将每一座桥用连接相应两个顶点的一条边来代替，从而得到一个图(如图 1(b)所示)。欧拉证明了这个问题没有解(详见本书 5.1 节)，并且推广了这个问题，给出了“对于一个给定的图，能否用某种方式走遍所有的边且没有重复”的判定法则。这项工作使欧拉成为图论及拓扑学的创始人。

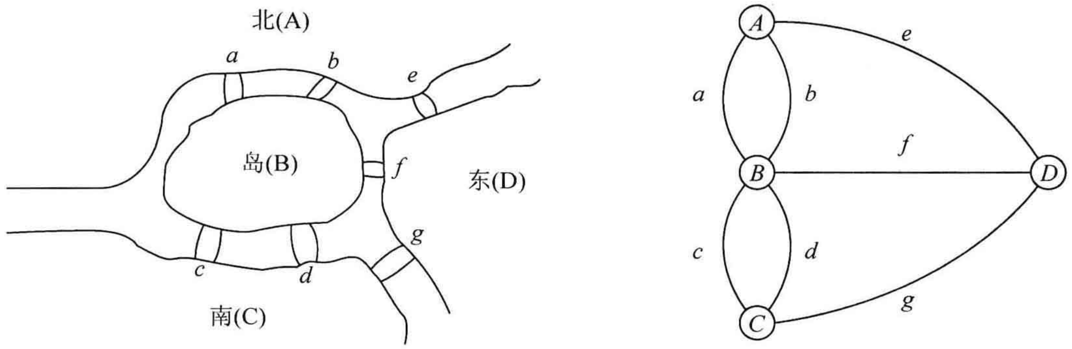
在此后的 200 多年时间里，图论的研究从萌芽阶段，逐渐发展成为数学的一个新分支。特别是从 20 世纪初期开始，在生产管理、交通运输、计算机和通信网络等方面涌现了许多离散性问题，这极大地促进了图论的发展。20 世纪 70 年代以后，由于高性能计算机的出现，使大规模的图论问题的求解成为可能。现在，图论理论广泛应用在运筹学、计算机科学、电子学、信息论、控制论、网络理论、经济管理等领域。

由于图论的重要性，越来越多的大学将图论单独作为一门课程来开设，把它作为数学、计算机科学、电子学、管理学等专业本科生和研究生的必修课或选修课。很多其他课程的内容也都涉及图论知识，如离散数学、运筹学、拓扑学等。介绍图论理论的教材逐渐增多，其中也不乏优秀的教材，如参考文献[1]～[7]。这些课程和教材或者是侧重于完整的图论知

---

① 本文中关于图论课程教学改革的一些思想，已经发表在《计算机教育》2009 年第 20 期上，论文题目为《计算机专业图论课程教学改革探索》，即参考文献[20](获得《计算机教育》杂志社举办的“英特尔杯”2009 年全国计算机教育优秀论文评比二等奖)。

识体系介绍以及复杂的图论定理的数学证明，或者是侧重于从应用数学的角度研究图论在各领域的应用。



(a) 布格河　　(b) 转化后的图

图 1　七桥问题

另外，为了实现用计算机程序求解各种应用问题，计算机科学家抽象出许多数据结构，如栈、队列、堆、树及二叉树、图等，其中图是最重要的数据结构之一，也是应用得最广的数据结构之一。数据结构课程是专门研究这些数据结构的描述、实现及应用的课程。数据结构课程讲到图论部分时，侧重于图结构的描述、图结构的存储、少量基本的图论算法的实现等。

许多学生(特别是计算机专业的学生)在学习图论时，都不满足于图论算法的手工和草稿纸演算，迫切地想知道如何用程序实现图论中的算法，以及如何将这些算法思想用来求解实际问题。据作者调查统计①，市面上侧重于用程序实现图论算法，并通过例题阐述图论算法思想及其应用的教材少之又少，本书希望能弥补这一缺憾。所以本书立足于图论算法理论和思想的描述及程序实现，并以大量的 ACM/ICPC 竞赛题目来阐述图论算法思想在求解这些题目中的应用。接下来简要地介绍 ACM/ICPC 程序设计竞赛。

二、ACM/ICPC 程序设计竞赛

1. ACM/ICPC

ACM/ICPC(ACM International Collegiate Programming Contest，国际大学生程序设计竞赛)是由美国计算机协会 ACM(Association for Computing Machinery)主办的，世界上公认的规模最大、水平最高的国际大学生程序设计竞赛，其目的旨在使大学生运用计算机来充分展示自己分析问题和解决问题的能力。该项竞赛从 1977 年第 1 次举办世界总决赛以来，至今已连续举办 30 多届了。该项竞赛一直受到国际各知名大学的重视，并受到全世界各著名计算机公司的高度关注。

ACM/ICPC 竞赛分区域预赛和总决赛两个阶段进行，各预赛区第 1 名自动获得参加世界总决赛的资格。世界总决赛安排在每年的 3～4 月举行，而区域预赛安排在上一年的 9～12 月在各大洲举行。

---

① 作者对互动出版网站(www.china-pub.com)和卓越亚马逊网站(www.amazon.cn)上列出的全部图论相关书目及目录进行了仔细的分析，从而得出的结论。

ACM/ICPC 竞赛以组队方式进行比赛，每支队伍由不超过 3 名队员组成，比赛时每支队伍只能使用一台计算机。在 5 个小时的比赛时间里，参赛队伍要解答 6～10 道指定的题目。排名时，首先根据解题数目来排名，如果多支队伍解题数量相同，则根据队伍的总用时进行排名(用时越少，排名越靠前)。每支队伍的总用时为每道解答正确的题目的用时总和。每道解答正确的题目的用时为从比赛开始计时到该题目解答被判定为正确的时间，其间每一次错误的提交运行将被加罚 20 分钟时间。最终未正确解答的题目不记入总时间，其提交也不加罚时间。

ACM/ICPC 竞赛在公平竞争的前提下，提供了一个让大学生充分展示用计算机解决问题的能力与才华的平台。ACM/ICPC 竞赛鼓励创造性和团队协作精神，鼓励在编写程序时的开拓与创新，它考验参赛选手在承受相当大的压力下所表现出来的非凡能力。竞赛所触发的大学生的竞争意识为加速培养计算机人才提供了最好的动力。竞赛中对解决问题的苛刻要求和标准使得大学生对解决问题的深度和广度展开最大程度的追求，也为计算机科学的研究和发展作了一个最好的导向。

由于图论有着丰富的算法和大量灵活的应用问题，所以一直以来图论题目在 ACM/ICPC 竞赛中都占了比较大的比重。图的遍历、活动网络、最小生成树、最短路径、图的可行遍性问题、网络流问题、匹配问题、图的连通性、图的着色等都有大量经典的题目，几乎涵盖了图论完整的知识体系。

2. 在线评判网站

随着 ACM/ICPC 程序设计竞赛的推广，各种程序在线评判(Online Judge，OJ)网站也应运而生，这为程序设计爱好者提供了一种新的程序实践方法：在线程序实践。

在线程序实践是指由 OJ 网站提供题目，用户在线提交程序，OJ 网站的在线评判系统实时评判并反馈评判结果。这些题目一般具有较强的趣味性和挑战性，评判过程和结果也公正及时，因此能引起用户的极大兴趣。

用户在解题时编写的解答程序通过网页提交给在线评判系统称为提交运行，每一次提交运行会被判为正确或者错误，判决结果会及时显示在网页上。

用户从评判系统收到的反馈信息包括以下几种。

"Accepted" ——程序通过评判！

"Compile Error" ——程序编译出错。

"Time Limit Exceeded" ——程序运行超过该题的时间上限还没有得到输出结果。

"Memory Limit Exceeded" ——内存使用量超过题目里规定的上限。

"Output Limit Exceeded" ——输出数据量过大(可能是因为陷入死循环了)。

"Presentation Error" ——输出格式不对，可检查空格、空行等细节。

"Run Time Error" ——程序运行过程中出现非正常中断，如数组越界等。

"Wrong Answer" ——用户程序的输出错误。

用户可以根据 OJ 系统反馈回来的评判结果反复修改程序，直到最终收获 Accept(程序正确)。这个过程不仅能培养用户独立分析问题、解决问题的能力，而且每成功解决一道题目都能给用户带来极大的成就感。

三、本书安排

本书共分 9 章，每章内容安排如下。

第 1 章介绍图的一些基本概念，以及图的两种重要存储表示方法：邻接矩阵和邻接表，并初步讨论存储方式对图论算法复杂度的影响。

第 2 章讨论图的遍历，遍历是很多图论算法的基础。本章介绍两种重要的遍历方法：深度优先搜索和广度优先搜索，并对这两种遍历算法的思想、程序实现、算法复杂度作详细的分析和讨论。本章还讨论活动网络，包括 AOV 网络与拓扑排序问题、AOE 网络与关键路径问题。

第 3 章讨论树与图的生成树，主要介绍求无向连通图最小生成树的 3 种算法：克鲁斯卡尔(Kruskal)算法、Boruvka 算法和普里姆(Prim)算法，并对这 3 种算法的思想、程序实现、算法复杂度作详细的分析和讨论。另外，本章还讨论判断生成树是否唯一的方法。

第 4 章讨论了有向网(或无向网)中一个典型的问题：最短路径问题。本章介绍求解最短路径问题的 4 种算法：Dijkstra 算法、Bellman-Ford 算法、SPFA 算法和 Floyd 算法，这 4 个算法分别适用于有向网(或无向网)中各边权值的取值的不同情形及问题求解的不同需要。本章着重对这 4 种算法的思想、递推过程、算法复杂度作详细的讨论，并对这 4 种算法作详细的对比分析。本章还介绍求最短路径的算法思想在求解差分约束系统中的应用。

第 5 章讨论可行遍性问题，包括欧拉回路、汉密尔顿回路以及中国邮递员问题。前两个概念容易混淆，欧拉回路要求经过每条边一次且仅一次并回到出发点，而汉密尔顿回路要求经过每个顶点一次且仅一次并回到出发点。本章介绍相关概念及定理，并讨论这两种回路及中国邮递员问题的求解方法和应用。

第 6 章讨论网络流问题。许多系统包含了流量问题，例如，公路系统中有车辆流，控制系统中有信息流，供水系统中有水流，金融系统中有现金流等。从问题求解的需求出发，网络流问题可以分为：网络最大流，流量有上下界的网络的最大流和最小流，最小费用最大流，流量有上下界的网络的最小费用最大流等。本章介绍各种网络流问题的求解方法。

第 7 章讨论点支配集、点覆盖集、点独立集、边覆盖集、边独立集(匹配)，这些概念之间存在一定的联系，也容易混淆。本章主要讨论各种匹配问题，以及求解二部图最大匹配的算法、程序实现和应用。

第 8 章讨论图的连通性，这是图论中一个重要的概念。本章介绍无向连通图和非连通图，无向图的点连通性(包括割顶集、割点、顶点连通度、点双连通图等)、边连通性(包括割边集、割边、边连通度、边双连通图等)、有向图的强连通性(包括强连通、弱连通和单连通)。本章着重介绍上述概念及求解算法。

第 9 章讨论平面图和着色问题。本章介绍平面图和非平面图的概念、平面图的判定方法，以及图的顶点着色、边着色、平面图的面着色等概念和求解算法。

四、本书读者对象及本书特点

本书的读者对象为计算机专业学生或对 ACM/ICPC 竞赛感兴趣的学生，可以作为高等院校计算机(或相关专业)的图论等相关课程的主教材，也可作为 ACM/ICPC 竞赛的辅导教材。学生或读者应该具备 C/C++语言知识，已经掌握了一定的程序设计思想和方法，具备

一定的算法分析与设计能力，并能熟练使用数据结构。

本书在内容取材、描述上具有以下特点。

(1) 许多图论教材对图论概念的描述不一致，造成读者的阅读困难，本书试图改变这一现状，在每个概念的表述上作者查阅了大量的图论著作并进行比较分析，作者对每个概念采用大多数图论教材采用(或约定)的名词、定义方法等。

(2) 本书对图论算法思想的描述尽可能采用浅显易懂的语言来描述。

(3) 本书忽略所有图论定理的证明，着重分析图论算法的思想，重点在于这些图论算法的程序实现。对图论算法的程序实现是以 ACM/ICPC 例题来阐述的，本书共收录了 130 余道 ACM/ICPC 竞赛题目，例题和练习题各约占一半。本书附录列出了本书所有例题和练习题在 ZOJ、POJ 上的题号。

(4) 本书图表内容丰富，共绘制了 270 余幅图表。

(5) 为方便读者阅读和使用，作者对本书中出现的图论术语、符号、图论算法及应用分别作了索引，列在本书后面。

五、关于本书例题和练习题的说明

本书例题和练习题占了比较多的篇幅，为了尽可能压缩本书的篇幅，在此对本书的例题和练习题有以下约定。

(1) 删除了所有例题代码中的头文件，为此在这里对头文件包含作一些说明。因为 ACM/ICPC 题目对程序运行时间要求很严格，所以本书除例 2.8 外所有例题的输入/输出均采用 C 语言的输入/输出方式(即 scanf 函数和 printf 函数)。本书例题代码需要包含的头文件一般为：

```
#include <cstdio>          //或#include <stdio.h>
#include <cstring>         //或#include <string.h>
```

如果使用了 STL 中的栈 stack(或队列 queue、向量 vector，双端队列 deque)，则还需要包含以下头文件，并使用命名空间。

```
#include <stack>           //或#include <queue>，或#include <vector>，或#include <deque>
using namespace std;       //使用命名空间
```

另外，根据不同的情况可能还需要包含以下头文件：

```
#include <algorithm>       //如果调用了排序函数 sort
#include <cstdlib>         //如果调用了排序函数 qsort
#include <cmath>           //如果调用了数学函数
```

例 2.8 中的代码用到了 string 类，所以只能采用 C++语言的输入/输出方式(即 cin 和 cout)，因此例 2.8 需要包含以下头文件：

```
#include <iostream>        //支持 C++输入/输出的头文件
#include <string>          //string 类头文件
#include <cstring>         //字符处理函数 strcpy、memset 等
using namespace std;       //命名空间
```

(2) 每道题目的样例输入/样例输出中一般只给出一个测试数据，除非在题目的分析中需要借助不同的测试数据来解释算法需要考虑的不同情形。

六、关于插图格式的说明

为了便于读者阅读和理解书中的插图，编者对书中的插图有以下格式约定。

(1) 顶点序号：顶点序号一般从 1 开始计起；但第 4、6 章的插图顶点序号一般从 0 开始计起，因为这两章的算法和应用问题都包含源点，顶点 0 一般作为源点。另外，个别例题和练习题中顶点序号是从 0 开始计起，本书遵守题目的规定，不作修改。

(2) 顶点的表示：本书统一用两种圆圈表示顶点，较小的空心圆圈和较大的空心圆圈；前者在必要的时候(如区分二部图两个顶点集合中的顶点)可以填充为实心圆圈；后者主要是考虑到很多情况下需要将顶点的序号或其他标识顶点的符号放在圆圈内，因为顶点旁可能有其他的参数(如第 6 章中顶点的标号、第 8 章中顶点的深度优先数等)，或者是为了突出顶点的重要性(比如无向图的连通分量等)。

七、致谢

本书收录了 130 余道 ACM/ICPC 竞赛题目，这些题目在阐述图论算法思想、演示图论算法应用等方面起着重要的作用，部分例题的解答程序也参考了网络上发布的一些源代码。同时，本书在编写过程中还参考了国内外多本优秀的图论教材(详见参考文献)。在此，编者对这些题目、源代码和图论教材的作者一并表示衷心的感谢！

本书的编写和出版得到了 2010 年浙江省教育科学规划研究课题“以大学生学科竞赛为契机推动课程群的规划与建设”(编号：SCG156)的支持，在此表示感谢！

由于作者水平有限，难免有疏漏之处，欢迎读者指正，或者读者有什么好的建议，都可以联系编者：w_guiping@163.com。不胜感激！

编　者

2010 年 10 月

目　录

第 1 章　图的基本概念及图的存储 1

1.1　基本概念 1

1.1.1　有向图与无向图 1

1.1.2　完全图、稀疏图、稠密图 2

1.1.3　顶点与顶点、顶点与边的关系 3

1.1.4　顶点的度数及度序列 3

1.1.5　二部图与完全二部图 5

1.1.6　图的同构 6

1.1.7　子图与生成树 6

1.1.8　路径 8

1.1.9　连通性 8

1.1.10　权值、有向网与无向网 10

1.2　图的存储表示 10

1.2.1　邻接矩阵 10

1.2.2　邻接表 17

1.2.3　关于邻接矩阵和邻接表的进一步讨论 24

练习 24

第 2 章　图的遍历与活动网络问题 25

2.1　DFS 遍历 25

2.1.1　DFS 算法思想 25

2.1.2　DFS 算法的实现及复杂度分析 26

2.1.3　例题解析 29

练习 38

2.2　BFS 遍历 41

2.2.1　BFS 算法思想 41

2.2.2　BFS 算法的实现及复杂度分析 42

2.2.3　关于 DFS 算法和 BFS 算法的说明 44

2.2.4　例题解析 45

练习 58

2.3　活动网络——AOV 网络 63

2.3.1　AOV 网络与拓扑排序 63

2.3.2　拓扑排序实现方法 65

2.3.3　关于拓扑排序的进一步说明 70

2.3.4　例题解析 71

练习 79

2.4　活动网络——AOE 网络 81

2.4.1　AOE 网络与关键路径 81

2.4.2　关键路径求解方法 82

第 3 章　树与图的生成树 88

3.1　树与森林 88

3.1.1　树 88

3.1.2　森林 88

3.2　生成树及最小生成树 89

3.2.1　生成树 89

3.2.2　最小生成树 89

3.3　克鲁斯卡尔(Kruskal)算法 90

3.3.1　Kruskal 算法思想 90

3.3.2　等价类与并查集 91

3.3.3　Kruskal 算法实现 95

3.3.4　Boruvka 算法 99

3.3.5　例题解析 99

练习 105

3.4　普里姆(Prim)算法 109

3.4.1　Prim 算法思想 109

3.4.2　Prim 算法实现 110

3.4.3　关于 Prim 算法的进一步讨论 114

3.4.4　例题解析 114

练习 119

3.5　判定最小生成树是否唯一 123

3.5.1　最小生成树不唯一的原因分析 123

3.5.2　判定最小生成树是否唯一的方法 124

3.5.3　例题解析 126

第4章 最短路径问题......131
4.1 边上权值非负情形的单源最短路径问题——Dijkstra 算法......131
4.1.1 算法思想......131
4.1.2 算法实现......133
4.1.3 关于 Dijkstra 算法的进一步讨论......137
4.1.4 例题解析......137
练习......144
4.2 边上权值为任意值的单源最短路径问题——Bellman-Ford 算法......148
4.2.1 算法思想......148
4.2.2 算法实现......150
4.2.3 关于 Bellman-Ford 算法的进一步讨论......153
4.2.4 例题解析......156
练习......164
4.3 Bellman-Ford 算法的改进——SPFA 算法......167
4.3.1 算法思想......167
4.3.2 算法实现......167
4.3.3 关于 SPFA 算法的进一步讨论......171
4.3.4 例题解析......171
练习......178
4.4 所有顶点之间的最短路径——Floyd 算法......180
4.4.1 算法思想......180
4.4.2 算法实现......182
4.4.3 关于 Floyd 算法的进一步分析......185
4.4.4 例题解析......185
练习......192
4.5 差分约束系统......198
4.5.1 差分约束系统与最短路径......198
4.5.2 例题解析......200
练习......208
第5章 可行遍性问题......212
5.1 欧拉回路......212
5.1.1 基本概念及定理......212
5.1.2 欧拉回路的判定......216
练习......223
5.2 欧拉回路的求解......223
5.2.1 DFS 搜索求解欧拉回路......223
5.2.2 Fleury(佛罗莱)算法......232
练习......236
5.3 中国邮递员问题......237
5.4 汉密尔顿回路......238
5.4.1 基本概念及定理......239
5.4.2 汉密尔顿回路求解......241
第6章 网络流问题......246
6.1 网络最大流......246
6.1.1 基本概念......247
6.1.2 最大流最小割定理......251
6.1.3 网络最大流的求解......252
6.1.4 一般增广路方法——Ford-Fulkerson 算法......253
6.1.5 最短增广路算法......261
6.1.6 连续最短增广路算法——Dinic 算法......264
6.1.7 一般预流推进算法......266
6.1.8 最高标号预流推进算法......270
6.1.9 网络最大流算法总结......270
6.1.10 例题解析......271
练习......285
6.2 最小割的求解......289
练习......301
6.3 流量有上下界的网络的最大流和最小流......304
6.3.1 流量有上下界的容量网络......304
6.3.2 流量有上下界的网络的最大流......307
6.3.3 流量有上下界的网络的最小流......307
6.3.4 例题解析......313
练习......325
6.4 最小费用最大流......327
6.4.1 基本概念......327
6.4.2 最小费用最大流算法......328

6.4.3　例题解析 330
练习 338

第 7 章　支配集、覆盖集、独立集与匹配 342

7.1　点支配集、点覆盖集、点独立集 342
7.1.1　点支配集 342
7.1.2　点覆盖集 344
7.1.3　点独立集 345
7.1.4　点支配集、点覆盖集、点独立集之间的联系 347
7.2　点支配集、点覆盖集、点独立集的求解 347
7.2.1　逻辑运算 347
7.2.2　极小点支配集的求解 348
7.2.3　极小点覆盖集、极大点独立集的求解 348
7.3　边覆盖集与边独立集 349
7.3.1　边覆盖集 349
7.3.2　边独立集(匹配) 350
7.3.3　最大边独立集(最大匹配)与最小边覆盖集之间的联系 352
7.4　匹配问题 353
7.4.1　完美匹配 353
7.4.2　二部图的完备匹配与完美匹配 354
7.4.3　最佳匹配 354
7.4.4　匹配问题求解的基本概念及思路 354
7.5　二部图最大匹配问题的求解 356
7.5.1　网络流解法 356
7.5.2　匈牙利算法 358
7.5.3　例题解析 361
练习 377

第 8 章　图的连通性问题 382

8.1　基本概念 382
8.1.1　连通图与非连通图 382
8.1.2　无向图的点连通性 383
8.1.3　无向图的边连通性 385
8.1.4　无向图顶点连通性和边连通性的联系 386
8.1.5　有向图的连通性 386
8.2　无向图点连通性的求解及应用 387
8.2.1　关节点的求解 387
8.2.2　重连通分量的求解 394
8.2.3　顶点连通度的求解 396
练习 401
8.3　无向图边连通性的求解及应用 403
8.3.1　割边的求解 403
8.3.2　边双连通分量的求解 407
8.3.3　边连通度的求解 414
练习 416
8.4　有向图强连通性的求解及应用 418
8.4.1　有向图强连通分量的求解算法 418
8.4.2　有向图强连通分量的应用 421
练习 435

第 9 章　平面图及图的着色问题 438

9.1　基本概念 438
9.1.1　平面图与非平面图 438
9.1.2　区域与边界 439
9.1.3　极大平面图与极小非平面图 440
9.1.4　平面图的对偶图 440
9.1.5　关于平面图的一些定理 441
9.2　欧拉公式及其应用 441
9.2.1　欧拉公式 441
9.2.2　欧拉公式的应用 442
练习 445
9.3　平面图的判定 446
9.4　图的着色问题 447
9.4.1　地图染色与四色猜想 447
9.4.2　图的着色 448
9.4.3　图着色的应用 450
9.4.4　图着色求解算法及例题解析 451
练习 455

附录　本书例题和练习题目录 457

索引 461

参考文献 469

第 1 章　图的基本概念及图的存储

图是一种重要的数学模型和数据结构，通常用来描述某些事物之间的某种特定关系，图可以对自然科学和社会科学中许多领域的问题进行恰当的描述或建模，因而在很多领域都有广泛的应用。本章介绍图论的一些基本概念及图的两种存储表示方法：邻接矩阵和邻接表。

1.1　基 本 概 念

图论知识的一个特点就是概念特别多，为了使读者尽快进入到图论算法层次，而不是停留在概念层面上，本节只介绍一些基本的概念，其他概念(如遍历、拓扑排序、网络流等)在其他章节里需要用的时候再补充介绍。

1.1.1　有向图与无向图

图(Graph)是由顶点集合和顶点间的二元关系集合(即边的集合或弧的集合)组成的数据结构，通常可以用 $G(V, E)$来表示。其中**顶点集合**(Vertext Set)和**边的集合**(Edge Set)分别用 $V(G)$和 $E(G)$表示。$V(G)$中的元素称为**顶点**(Vertex)，用 u、v 等符号表示；顶点个数称为图的**阶**(Order)，通常用 n 表示。$E(G)$中的元素称为**边**(Edge)，用 e 等符号表示；边的个数称为图的**边数**(Size)，通常用 m 表示。

例如，图 1.1(a)所示的图可以表示为 $G_1(V, E)$。其中，顶点集合 $V(G_1) = \{ 1, 2, 3, 4, 5, 6 \}$，集合中的元素为顶点(用序号代表，在其他图中，顶点集合中的元素也可以是其他标识顶点的符号，如字母 A、B、C 等)；边的集合为：

$$E(G_1) = \{ (1, 2), (1, 3), (2, 3), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5), (4, 5) \}。$$

在上述边的集合中，每个元素(u, v)为一对顶点构成的无序对(用圆括号括起来)，表示与顶点 u 和 v **相关联**的一条**无向边**(Undirected Edge)，这条边没有特定的方向，因此(u, v)与(v, u)是同一条的边。如果图中所有的边都没有方向性，这种图称为**无向图**(Undirected Graph)。

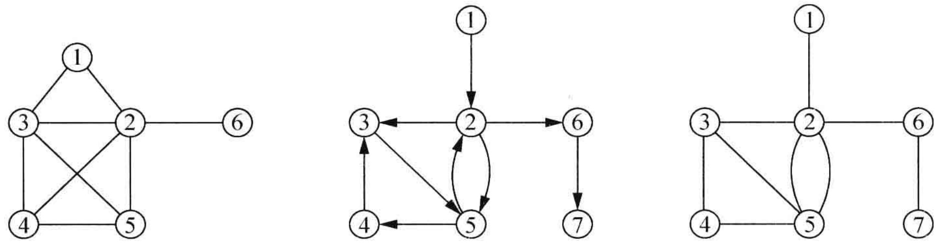
图 1.1(b)所示的图可以表示为 $G_2(V, E)$，其中顶点集合 $V(G_2) = \{ 1, 2, 3, 4, 5, 6, 7 \}$，集合中的元素也为顶点的序号；边的集合为：

$$E(G_2) = \{ <1, 2>, <2, 3>, <2, 5>, <2, 6>, <3, 5>, <4, 3>, <5, 2>, <5, 4>, <6, 7> \}。$$

在上述边的集合中，每个元素$<u, v>$为一对顶点构成的有序对(用尖括号括起来)，表示从顶点 u 到顶点 v 的**有向边**(Directed Edge)，其中 u 是这条有向边的**起始顶点**(Start Vertex)，简称**起点**，v 是这条有向边的**终止顶点**(End Vertex)，简称**终点**，这条边有特定的方向，由 u 指向 v，因此$<u, v>$与$<v, u>$是两条不同的边。例如，在图 1.1(b)有向图 G_2 中，$<2, 5>$和$<5, 2>$是两条不同的边。如果图中所有的边都是有方向性的，这种图称为**有向图**(Directed Graph)。

有向图中的边也可以称为**弧**(Arc)。有向图也可以表示成 $D(V, A)$，其中 A 为弧的集合。

有向图的**基图**(Ground Graph)：忽略有向图所有边的方向，得到的无向图称为该有向图的基图。例如，图 1.1(c)所示为图 1.1(b)中有向图 G_2 的基图。



(a) 无向图G_1　　(b) 有向图G_2　　(c) 有向图G_2的基图

图 1.1　无向图与有向图

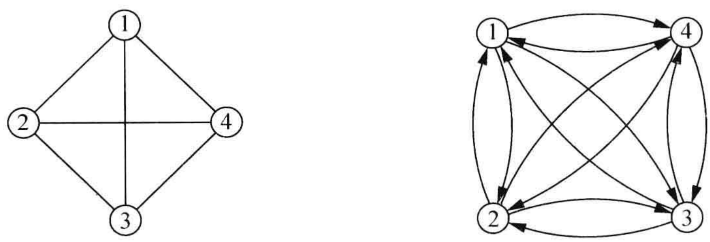
说明：如果一个图中某些边具有方向性，而其他边没有方向性，这种图可以称为**混合图**(Mixed Graph)；如无特殊说明，本书的讨论仅限于无向图和有向图，不包括混合图。

1.1.2　完全图、稀疏图、稠密图

许多图论算法的复杂度都与图中顶点个数 n 或边的数目 m 有关，甚至 m 与 $n\times(n-1)$之间的相对关系也会影响图论算法的选择。下面介绍几个与顶点个数、边的数目相关的概念。

完全图(Complete Graph)：如果无向图中任何一对顶点之间都有一条边，这种无向图称为完全图。在完全图中，阶数和边数存在关系式：$m = n\times(n-1)/2$。例如，图 1.2(a)所示的无向图就是完全图。阶为 n 的完全图用 K_n 表示。例如，图 1.2(a)所示的完全图为 4 阶完全图 K_4。

有向完全图(Directed Complete Graph)：如果有向图中任何一对顶点 u 和 v，都存在$<u, v>$和$<v, u>$两条有向边，这种有向图称为有向完全图。在有向完全图中，阶数和边数存在关系式：$m = n\times(n-1)$。例如，图 1.2(b)所示的有向图就是有向完全图。



(a) 完全图K_4　　(b) 有向完全图

图 1.2　完全图与有向完全图

稀疏图(Sparse Graph)：边或弧的数目相对较少(远小于 $n\times(n-1)$)的图称为稀疏图。有的文献认为，边或弧的数目 $m<n\log(n)$的无向图或有向图，称为稀疏图。例如，图 1.3(a)所示的无向图可以称为稀疏图。

稠密图(Dense Graph)：边或弧的数目相对较多的图(接近于完全图或有向完全图)称为稠密图。例如，图 1.3(b)所示的无向图可以称为稠密图。