国外经典教材

PEARSON
Addison Wesley

C++算法（第3版）

——图算法

Algorithms in C++

Third Edition

Part 5

Graph Algorithms

（美） Robert Sedgewick 著

林 琪 译

Pearson Education

清华大学出版社

国外经典教材

C ++ 算 法 (第3版)

——图 算 法

(美) Robert Sedgewick 著
林 琪 译

清华大学出版社
北 京

内 容 简 介

本书所关注的是图算法领域。从实用的视角，以独特的结构将有关内容组织在一起，从而使读者不仅可以对这一领域有系统性的认识，而且还可在实践中灵活使用所提供的算法工具。本版中，增加了数以千计的新练习、数百个新图表以及数十个新程序，而且对所有的图表和程序都做了详尽的注释说明；不仅涵盖了新的主题，还对许多经典算法提供了更为充分的解释。所有读者都可从中得到极为丰富的学习资料，从而更好地理解基本概念。

本书以C++作为算法描述语言，易于理解、便于应用。可作高校计算机相关专业本科生和研究生的教材和补充读物，也可供相关领域工程技术人员参考。

图书在版编目（CIP）数据

C++算法（第3版）——图算法/（美）塞吉维克著；林琪译.—北京：清华大学出版社，2003.9
（国外经典教材）
书名原文： Algorithms in C++ (Third Edition), Part 5: Graph Algorithms
ISBN 7-302-07251-5
Ⅰ.C… Ⅱ.①塞… ②林… Ⅲ.C语言—程序设计—数值计算 Ⅳ.TP312

中国版本图书馆CIP数据核字（2003）第082878号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969

文稿编辑：葛昊晗
封面设计：立日新设计公司
印 刷 者：北京彩艺印刷有限公司
装 订 者：三河市李旗庄少明装订厂
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：26　**字数**：630千字
版　　次：2003年10月第1版　2003年10月第1次印刷
书　　号：ISBN 7-302-07251-5/TP·5268
印　　数：1～4000
定　　价：49.00元

前　言

图和图算法在当今的计算应用中颇为常见。对于在实际中出现的图处理问题，本书描述了一些已知的最重要的解决方法。由于需要相关知识的人日渐增多，这本书的主要目的就是让他们了解这些方法及其所蕴藏的基本原则。全书由最基本的原则展开，并从基本概念开始介绍，逐步过渡到经典方法，最后对仍在开发中的最新技术加以讨论。在对算法和应用的描述中，我们提供了精心挑选的示例、详尽的图表以及完备的补充说明。

算法

《C++算法》研究当前所使用的最为重要的计算机算法本书是其中的图算法卷。

在学习计算机科学课程之初，即学生已经掌握了基本的编程技巧，熟悉计算机系统，但是尚未选修计算机科学或计算机应用高级领域中的专业课程时，将本书作为教材是很有用的。本书也可用于自学，对从事计算机系统或应用程序开发的人来说，将本书用作参考书也是相当有用的，书中包含了实用算法的实现，并对这些算法的性能特性提供了详尽的信息。本书适于作为这一领域的入门读物。

多年以来，《C++算法》一书已经得到了世界各地的学生和程序员的广泛使用，在第3版中，我完全重写了有关内容，并且增加了数千个新练习、数百个新图表以及数十个新程序，而且对所有的图表和程序做了详尽的注释说明。在此不仅涵盖了新的主题，而且还对许多经典算法提供了更为充分的解释。全书强调了抽象数据类型，从而使得有关程序的应用面更广，而且与当今的面向对象编程环境也更为相关。对于已经阅读过本书以前版本的人来说，会从这一版中找到相当多的新内容；而对于所有读者而言，都能从中得到极为丰富的学习资料，可以更好地理解基本概念。

这些书不仅适合程序员和计算机科学专业的学生阅读。每一个使用计算机的人都希望它能运行得更快，或者可解决更大规模的问题。我们所考虑的算法正代表着近50年发展起来的知识体系，该体系是在各种各样的应用中有效地使用计算机的基础。从物理学中的多体仿真问题到分子生物学中的基因序列问题，在此所描述的基本方法在科学研究中已日显重要；另外，对于从数据库系统到 Internet 搜索引擎等当今的软件系统，这些基本方法也已经成为其基本的组成部分。随着计算机应用的覆盖面越来越广，基本算法的影响也日益显著，特别是本书所介绍的基本图算法，作用更为突出。广大学生以及专业人士可能会参与完成各种计算机应用，随着这些应用中相关需求的增长，本书的目标就是要提供一个有效的资源，从而使他们充分了解并明智地使用图算法。

本书范围

本书共包括 6 章，分别介绍图的属性和类型、图搜索、有向图、最小生成树、最短

路径以及网。这些描述的目的是为了使读者能够了解尽可能多的基本图算法，并对其基本属性有所理解。

如果你曾经学过有关算法设计和分析基本原则的课程，并且有利用诸如 C++，Java 或 C 等高级语言编程的经验，那么对于在此介绍的内容，就会充分领略到它的价值。本书假设你已经对数组、链表以及 ADT(Abstract Data Type，抽象数据类型)设计等有了基本的了解，而且使用过优先队列、符号表以及并查 ADT。

图和图算法的基本属性由最基本的原则即可建立，但要充分理解，则往往需要拥有博大精深的数学背景。在此对高级数学概念的讨论很简短，而且是概括性和描述性的，要想对图算法有更深入的认识，应该有更高的数学水平。不过，数学水平各不相同的读者都可从此书中获益。这种说法可做如下考虑：相对于并非任何人都能理解的一些高级算法，每个人都应该理解并使用的基本图算法只是略有差异。在此的主要意图是结合贯穿于全书的其他方法来讨论重要的算法，而不是对所有数学知识做全面的介绍。不过，好的数学基础往往要求严格的行事方式，而这通常可使我们得到好的程序，因此我尽量在理论家所崇尚的形式规范性和实践家所需要的内容丰富性之间进行权衡，同时也不损害严格性。

教学使用

在本书的讲授方式上存在很大的灵活性，这取决于教师的偏好，同时也依赖于学生所做的准备。可以看到，这里所描述的算法多年以来已经得到了广泛使用，而且无论是对实际工作的程序员还是计算机科学专业的学生而言，这些算法都代表了基本的知识体系。本书可用于数据结构和算法的课程，因为它阐述了足够的基本内容；也可用于图算法的课程，这本书不仅足够详细，而且涵盖了高级的内容。有些教师可能希望强调与实现和实用有关的内容，而另外一些教师则可能希望把重点放在分析和理论概念上。

可将本书与我写的其他 C++算法书结合起来，作为一门更为全面的课程讲授。这样，教师就可以完全用一种一致的风格来介绍基础知识、数据结构、排序、查找和图算法等全部内容。通过访问本书主页，我们可以找到一套用于讲解的幻灯片、程序设计作业示例、为学生提供的交互式练习以及其他一些教学资料。

书中的练习 (几乎全都是在这一版中新增加的) 可分为多种类型。有一些是为了检查对正文中内容的理解，只要求读者完成某个示例，或者应用正文中所描述的概念。另外一些则涉及实现和整理算法，或者要做实验研究，从而对不同算法加以比较以了解其属性。还有一些练习则相当于知识储备，是对一些重要信息所做的相当详细的说明，而这些信息本身不适于放在正文里。阅读这些练习并加以思考，会使每个读者都有意想不到的收获。

实用算法

任何人若希望更为有效地使用计算机，都可以将这本书作为参考，或用于自学。有编程经验的人可以从书中找到有关一些特定主题的信息。一般地，你可以抽取书中的各

章独立地阅读。不过，有些情况下，某一章中的算法可能会用到前一章中所介绍的方法。

本书的定位是对很可能会在实际中使用的算法加以研究。本书对所讨论的工具 (即算法) 提供了详尽的信息，读者可以放心地实现和调试，并用这些算法来解决问题，或在应用中利用它们来提供有关功能。在此对所讨论的方法提供了完整的实现，同时，针对书中一系列一致的示例程序的操作做了描述。由于我们采用了实际代码，而不是编写伪代码，因此这些程序很快就可以在实际中使用。通过访问本书的主页可以得到程序的代码清单。

实际上，由算法的一个实际应用已经得到了本书中数百个图表。许多算法正是通过这些图表所提供的视觉维度直观地发现和得到的。

本书将详细讨论这些算法的特性以及它们可能在哪些情况下是有用的。在此可建立算法分析与理论计算机科学之间的联系。在适当的情况下，都将给出经验性的结果以及分析结果，从而说明为什么某些算法更为适用。如果有意义，还会对所讨论的实际算法与纯理论结果之间的关系加以描述。对于算法和实现的性能特性的特定信息，全书将对其进行综合性和概要性的讨论。

编程语言

书中所有实现所用的编程语言均为 C++。程序中使用了大量的标准 C++习惯用法，而且对于每个构造，正文中都做了简洁的描述。

Chris Van Wyk 和本人基于类、模板和重载操作符建立了一种 C++编程的风格，并认为这是一个将算法和数据结构表示为实际程序的有效方法。我们在实现的优雅性、简洁性、有效性和可移植性方面做了很大的努力。程序风格会尽可能保持一致，因此类似的程序看上去也是相似的。

本书的目标是以尽可能简单明了的方式来展示算法。对于许多算法而言，尽管所用的语言不同，但存在着相似性。作为一个突出的例子，Dijkstra 算法就是 Dijkstra 算法，无论采用 Algol-60，Basic，Fortran，Smalltalk，Ada，Pascal，C，C++，Modula-3，PostScript，Java 还是任何一种其他的编程语言(这样的语言可谓不计其数)来编写，也不管所在的是何种环境，均可以证实为有效的图处理方法。一方面，采用这些语言(以及其他多种语言)来实现算法会获得一些经验(本书的 C 版本已经面世，而 Java 版本不久也会推出)，代码会受到这些经验的影响；另一方面，对于这其中的一些语言，其属性会受其设计人员的经验所左右，而这些经验又来自于他对本书所讨论的部分算法和数据结构的使用。最后，我们认为本书所提供的代码不仅准确地定义了算法，而且在实际工作中也相当有用。

致谢

对于本书以前的版本，很多人都提供了有益的反馈。特别要指出的是，普林斯顿大学和布朗大学的数千学生多年以来承受了初稿的粗糙。特别要感谢 Trina Avery 和 Tom Freeman 对于第一版的出版所提供的帮助；还要感谢 Janet Incerpi，是她的创造性和聪明才智使我们有了最初的 (同时也是原始的) 数字计算机化排版硬件和软件，由此才产生

了第一版；要感谢 Marc Brown，他参与了算法可视化研究，而本书中的图表正来源于此；感谢 Dave Hanson 和 Andrew Appel，对于我有关编程语言的问题，他们总是乐于作答；感谢 Kevin Wayne，他耐心地回答了我所提出的关于网的基本问题。Kevin 力劝我在书中加入网的单纯形算法 (simplex algorithm)，而我直到看到 Dagstuhl 的 Ulrich Lauther 所做的介绍，才采纳了他的建议，认为这样做是可能的，本书第 22 章的实现所基于的正是 Ulrich Lauther 的思想。我还要感谢许多读者，他们为各个版本提供了详尽的评论，这其中包括：Guy Almes，Jon Bentley，Marc Brown，JayGischer，AllanHeydon，Kennedy Lemke，Udi Manber，Dana Richards，John Reif，M. Rosenfeld，Stephen Seidman，Michael Quinn 和 William Ward。

为了完成此第 3 版，我很荣幸地与 Addison-Wesley 的 Peter Gordon 和 Helen Goldstein 共事，尽管项目已经由一般性的修改转变为大幅度的重写，他们仍耐心地指导了整个项目的进行。我也很有幸能与 Addison-Wesley 的许多专业人员一起工作。这个项目的性质使得这本书对于其中许多人来说都是不同寻常的挑战，对于他们的忍耐力我致以诚挚的谢意。特别要感谢 Marilyn Rash，是他卓越的工作才使这本书能够在如此紧张的进度下顺利完成。

在写这本书的过程中，我得到了三位良师益友，在此要特别向他们表示感谢。首先，Steve Summit 从技术角度对各版本的初稿都做了认真仔细的检查，并向我提出了数千条详细的意见，尤以对程序的意见为最。Steve 很清楚我的目标是要提供优雅、高效而有用的实现，他的意见不仅有助于我对实现的一致性加以度量，而且还帮助我对其中一部分做了重大改进。其次，Lyn Dupré 也对初稿提出了数千条的详细意见，对我来说它们真是无价之宝，它们不仅帮助我纠正和避免了许多语法错误，而且更重要的是，使我发现了一种一致而连贯的编写风格，由此才使我能够将如此繁多的技术资料合理地整理在一起。第三，Chris Van Wyk 用 C++实现了我的所有算法，并做了调试，还回答了大量有关 C++的问题，是他帮助我建立了一种恰当的 C++编程风格，也是他将此初稿仔细地阅读了两遍。当我把 Chris 的许多 C++程序摒弃在一边时，他总是耐心地表示同意，而当我对 C++有了越来越多的了解之后，又不得不将他原来所写的程序放回来。能够有机会向 Steve，Lyn 和 Chris 学习，我确实无比感激，他们的作用对于这本书的完成至关重要。

我在这里所写的许多内容都受益于 Don Knuth 的授课和著作，他是我在斯坦福大学的导师。虽然 Don 对于这本书没有直接的影响，但在本书中仍然可以感受到他的存在，因为正是他为算法研究奠定了坚实的科学基础，从而使得像本书这样的工作才有可能完成。我的朋友兼同事 Philippe Flajolet 对于这本书也起到了类似的影响，算法分析得以发展为一个成熟的研究领域，他在其中也堪称主力。

非常感谢普林斯顿大学、布朗大学以及法国国立计算机与自动化研究所(Institut National de Recherce en Informatique et Automatique，INRIA)提供的支持，在这些地方我完成了本书的大部分工作；还要感谢美国国防部防御分析研究所 (Institute for Defense Analyses) 以及施乐的帕洛阿尔托研究中心 (Xerox Palo Alto Research Center)，在对它们进行访问期间，我也做了本书的一些工作。这本书的某些部分离不开国家自然科学基金 (National Science Foundation) 和海军研究中心 (Office of Naval Research) 的慷慨支持。

最后，我要感谢 Bill Bowen，Aaron Lemonick 和 Neil Rudenstine，他们为普林斯顿大学建立一个良好的学术环境做了许多工作，正是有此环境，我才能够在承担众多其他事务的同时完成本书的准备。

Robert Sedgewick
Marly-le-Roi，法国，1983
普林斯顿，新泽西州，1990
詹姆斯镇，罗得岛，2001

C++顾问所撰写的前言

为了用清晰自然的程序来实现图算法，Bob Sedgewick 和我多次编写了大部分程序。由于图的种类非常之多，而且有关它们的各种问题更是数不胜数，因此，我们最初即达成共识，不在全书中使用单一的类模式。需要注意，我们最终只用了两种模式：一种是1~3 章所用的简单模式，其中，图中的边要么存在要么不存在；另一种方法类似于第 4~6 章中的 STL (Standard Template Library，标准模板库) 容器，其中有更多信息与边相关。

用 C++类表示图算法有许多优点。对于图的有用函数，我们用类来收集其中通用的一部分 (如输入/输出) 。第 2 章用类分解出不同图搜索方法所共有的一些操作。在整本书中，对于从某个顶点发出的边我们用了一个迭代器类，这样无论图是如何存储的，程序都能工作。最重要的是，我们将图算法打包在类中，其构造函数对图加以处理，而成员函数则使我们可以访问所发现的结果。对于图算法来说，这种组织可使之很容易地将其他图算法作为子例程来使用，例如，程序 3.13 (基于强分量的传递闭包) 、程序 4.8 (Kruskal 的最小生成树算法)、程序 5.4 (基于 Dijkstra 算法的全源最短路径) 和程序 5.6 (有向无环图的最长路径)。这种趋势到了第 6 章可谓达到极致，在此大多数程序都建立在更高层次的抽象之上，并使用了本书前面所定义的类。

为了与我写的其他 C++算法图书做到一致，我们的程序主要依赖其中所定义的栈和队列类，而且用了两种低层实现来编写单链表的外部指针操作。本书特有的网格是：构造函数使用的是初始化而非赋值，另外我们使用了 STL 向量而不是数组。以下是对程序中所用 STL 向量函数的总结：

- 默认的构造函数创建一个空向量。
- 构造函数 `vec(n)` 创建一个有 `n` 个元素的向量。
- 构造函数 `vec(n,x)` 创建一个有 `n` 个元素的向量，每个元素均初始化为值 `x`。
- 成员函数 `vec.assign(n,x)` 置 `vec` 为一个有 `n` 个元素的向量，且每个元素均初始化为值 `x`。
- 成员函数 `vec.resize(n)` 对 `vec` 加以扩展或缩小，使其容量为 `n`。
- 成员函数 `vec.resize(n,x)` 对 `vec` 加以扩展或缩小，使其容量为 `n`，并将所有新元素初始化为值 `x`。

STL 还定义了使向量成为首类对象所需的赋值操作符、复制构造函数以及析构函数。

开始编写这些程序之前，我阅读了许多算法的非正式描述以及伪代码，不过仅实现了其中的一部分。我发现要把算法转换为实用程序，设计出所需的细节很有帮助，而且看到它们如期实现也是很有意思的一件事。我希望通过阅读和运行本书中的程序，也能帮助你更好地理解算法。

感谢 Jon Bentley，Brian Kernighan 和 Tom Szymanski，从他们那里我学到了许多关

于编程的知识；还要感谢 Debbie Lafferty，是他问我是否对这个项目有兴趣；另外还要感谢德鲁大学、朗讯科技和普林斯顿大学在学术上的支持。

Christopher Van Wyk
查塔姆，新泽西州，2001

关于练习的说明

对练习进行分类是一件相当困难的工作，因为对类似于这本书的读者来说，其知识水平和经验往往各不相同。不过，提供一些指导还是有益的，因此许多练习带有如下 4 种标记中的一种，以此帮助你确定如何对其进行处理。

用于检查对内容的理解的练习标记了一个空心三角形，如下所示：

△ **2.34** 请考虑图

3-7 1-4 7-8 0-5 5-2 3-8 2-9 0-6 4-9 2-6 6-4。

画出其 DFS 树，并根据此树找到图的桥及边连通分量。

更常见的情况是，这种练习直接与正文中的例子相关。它们的难度不大，但完成这些练习可能会使你了解到某个事实或掌握某个概念，而你在阅读正文时可能对此未加注意。

如果练习用于为内容增加新的信息或需要思考的信息，则标记了一个空心圆，如下所示：

○ **3.106** 请编写一个程序，对给定 DAG 可能的拓扑排序结果进行计数。

这种练习可促使你对与正文内容相关的重要概念多加思考，或者使你能够解答在阅读正文时所存在的疑问。即便是没有时间来完成，你也会发现仅仅阅读这些练习也是大有裨益的。

有一定难度的练习则标记了一个黑点，如下所示：

● **4.73** 如果一个图过大，在内存中一次只能存放其中的 V 条边，请描述如何找到这种图的 MST。

这种练习可能需要花费相当多的时间才能完成，这取决于你的经验。一般情况下，效率最高的方法是分多次来完成。

有些练习相当难(相对于大多数其他练习而言)，这些练习将标记两个黑点，如下所示：

●● **4.37** 请设计一个合理的生成器，用以生成有 V 个顶点和 E 条边的随机图，从而对于 Dijkstra 算法，使其基于堆的 PFS 实现的运行时间是超线性的。

这些练习类似于在研究文献中所解决的问题，不过本书中的内容可以使你做好准备，并愿意尝试解决这些问题(而且很可能会成功)。

这些标记力图做到与你的编程能力和数学能力无关。对于需要在编程或数学分析方面有一定基础的练习，则是不言而喻的。建议所有读者都通过实现算法来检验自己对这些算法的理解。不过，尽管对于一个实际工作的程序员或正在上编程课程的学生来说，

完成以下的练习可能很简单，但对于近期未编程的人来说则可能需要做大量的工作：

●● **1.74** 请编写一个程序，在平面上生成 V 个随机点，再建立一个网，其中用边(双向)将给定距离 d 内的各对顶点相互连接起来(请参见程序 3.20)，将每条边的权值置为它所连接的两个顶点之间的距离。请确定如何设置 d 从而使边的数目为 E。

同样的道理，对于算法属性的知识，建议所有读者都要对其分析基础进行研读。不过，尽管如下的练习对于一个科学家或正在上离散数学课程的学生来说可能很简单，但对于最近未从事数学分析的人来讲，则需要做大量的工作：

○ **3.5** 对于有 V 个顶点及 E 条边的无向图，存在多少个与之对应的有向图？

这里的练习确实太多，你不可能全部阅读它们并消化；我所希望的是，通过做足够的练习，可以促使你对感兴趣的主题有更宽泛的理解，而不仅仅是掌握通过阅读正文所得到的知识。

目　录

第 1 章　图的属性和类型

现实中许多计算应用都不仅仅只涉及到元素集，还会涉及到元素与元素之间的连接集。由这些连接所表示的关系自然而然地会引出大量问题：有没有办法沿着这些连接由一个元素达到另一个元素？由一个给定的元素可以到达多少个其他元素？何为由某个元素到达另一个元素的最佳路径？

为了对这些情况建立模型，我们使用了一种称为图(graph)的抽象对象。本章将详细分析图的基本属性，这为学习各种有助于回答前面所提出的问题的算法打下了基础。这些算法充分利用了我们从第Ⅰ～Ⅳ部分所讨论的计算工具。另外它们还可为解决重要应用中存在的问题奠定基础，如果没有好的算法技术，我们甚至对这些应用的解决方案无从考虑。

图论是组合数学中的一个主要分支，数百年来，对此已经做了深入的研究。已经证实图有许多重要而有用的属性，但仍有大量难题未被攻破。尽管需要学习的内容尚有很多，但在本书中我们将只是从有关图的众多知识中抽取出需要理解的部分，并使用到大量有用而基本的算法。

与我们所研究的许多其他问题领域类似，对于图的算法研究也相对较新。尽管一些基本算法很古老，但大多数有趣的算法都是在最近几十年发现的。即使是根据最简单的图算法，也会得到很有用的计算机程序，而我们将要分析的“非凡”算法当属目前已知最优雅也最有趣的一类。

为了阐明涉及到图处理的应用的多样性，我们首先考虑几个例子，由此对这个领域中的算法展开研究，该领域中的内容相当丰富。

地图　一个计划去旅行的人可能需要回答一些问题，诸如“从普林斯顿到圣琼斯，哪条路线花费最少?”。如果一个人更关注时间而不是花费，那么可能需要了解“从普林斯顿到圣琼斯，哪一条是最快的路线？”。为了回答这些问题，我们要对元素(城镇)之间有关连接(旅行路线)的信息进行处理。

超文本　在浏览 Web 时，我们会遇到一些文档，其中包括有对其他文档的引用(链接)，通过单击这些链接可以由一个文档转至另一个文档。整个 Web 即为一个图，这里的元素就是文档，而连接即为链接。搜索引擎可帮助我们在 Web 上查找信息，而图处理算法就是搜索引擎的基本组成部分。

电路　电路中包括诸如晶体管、电阻器和电容等元件，它们通过导线杂乱地相互连接。我们用计算机来控制生成电路的机器，并查看电路是否完成了所需的功能。我们要回答一些简单问题，如“这是一条最短路径吗？”，还可能要回答诸如“能否将此电路做在芯片上而不出现任何线路交叉”等复杂问题。在这种情况下，对于第一个问题的答案仅取决于连接(导线)的属性，而第二个问题的答案则需要有关导线、导线所连接元素以及芯片的自然约束等更详细的信息。

调度　制造过程中需要完成许多任务，而且还存在一组约束，其中指定某些任务在另外一些任务完成前不得开始。我们将约束表示为任务(元素)之间的连接，这样就面临着一

个经典的调度问题，即如何合理调度任务，从而既满足给定约束，同时又在最短时间内完成整个过程。

事务 某电话公司维护一个电话呼叫通信量的数据库。这里的连接表示为电话呼叫。对于其实际互联结构我们很感兴趣，因为我们需要布线并设置交换机从而高效地处理通信。另外还有一个例子，某财政机构要跟踪市场上的购/销订单状况。这里的连接表示两个客户之间的现金转移。若对这种情况下的连接结构有所了解，则会加强对市场实际情况的了解。

匹配 学生们要在诸如社会团体、大学或医学院等选择性机构应聘职位。这里的元素对应为学生和机构；连接则对应为应聘关系。我们希望找出某种方法，从而将感兴趣的学生与现有职位相匹配。

网络 计算机网络由互联的站点组成，这些站点可发送、转发和接收不同类型的消息。我们所关心的不仅仅是知道从每个站点到各个其他站点有可能得到某个消息，而且随着网络的调整，还对每两个站点之间互连性的维护有所关注。例如，我们可能希望对一个指定网络进行检查，从而确保其中不存在某一部分站点或连接处于太过“要害”的地位，以至于它们一旦出现问题将导致其余站点无法连接。

程序结构 编译程序通过构建图来表示大型软件系统中的调用结构。这里的元素即为组成系统的各个函数或模块；连接要么与一个函数调用另一个函数的可能性相关(静态分析)，要么与系统正在运行时的实际调用有关(动态分析)。我们需要对此图进行分析，从而确定如何最佳地为程序分配资源以做到最为高效。

从以上例子可以看出，在许多应用中，图都是最合适的抽象工具，同时在使用图时也可能会遇到很多计算问题。这些问题即为本书的重点。以上应用付诸现实时，所涉及的数据量着实非常大，而高效的算法则会影响到一个解决方案是否真正可行。

在我写的其他C++算法书中已经对图进行了简要介绍，实际上，并查算法就是图算法的一个基本例子。还用图曾说明了二维数组和链表的应用，用图说明了递归程序和基本数据结构的关系。所有链式数据结构都可表示为图，而我们所熟知的一些用于处理树和其他链式结构的算法都是图算法的特例。本章的目的就是为理解图算法提供一个背景环境，从简单算法到本书的复杂算法都包括在图算法范畴之内。

毫无例外，我们很想知道解决一个特定问题时哪种算法最为高效。研究图算法的性能特点具有很大的挑战性，其原因在于：

- 算法的开销不仅取决于元素集的属性，还依赖于连接集的很多属性(以及由连接所包含的图的全局属性)。
- 对于我们可能遇到的各种图类型，要为其建立准确的模型是相当困难的。

对于图算法，我们通常采用最坏情况的性能下限，即使在多数情况下，其实际性能表现得可能并非如此低下。幸运的是，正如我们将要看到的，不少算法都是最优的，其中仅涉及到少量不必要的工作。对于给定规模的所有图，其他算法消耗的资源是相同的。我们可以准确预测出这些算法在特定情况下是如何完成的。如果无法做出这样准确的预测，那么对于实际应用中可能遇到的不同类型的图，就需要特别注意其属性，而且必须就这些属性对于算法的性能可能会有何种影响做出评判。

我们将由图的基本定义和图的属性入手开始讨论，在此将介绍用于描述图的标准术

语。随后，我们定义了基本 ADT(抽象数据类型)接口，图算法的研究即利用此接口展开，另外还介绍了两种最重要的数据结构来表示图，即邻接矩阵表示和邻接表表示，此外还介绍了实现基本 ADT 函数的各种方法。接下来，我们将考虑一些能够生成随机图的客户程序，它们可用于检查算法，并有助于学习图的属性。所有这些内容为引入图处理算法奠定了基础，对于在图中查找路径，这些算法可以解决与之相关的三个经典问题，这说明了尽管图问题看上去可能差别不大，但其难度可能会大相径庭。本章最后，将回顾书中分析过的最重要的图处理问题，并根据其解决难度的不同放到了不同位置。

1.1　术语

与图相关的术语相当多。大多数术语都有简明的定义，为了便于参考，我们将在这里集中介绍这些术语。在介绍第 I 部分中的基本算法时，我们已经用过其中一些概念；而另外一些只有在介绍第 2～6 章的相关高级算法时才会涉及到。

定义 1.1　图由一个顶点集和一个边集组成，其中边将一对不同顶点连接在一起(而且至多只能有一条连接了某一对顶点的边)。

在一个有 V 个顶点的图中，用 0 到 V-1 作为各顶点的名字。采用这种命名机制的主要原因在于，我们可以利用向量索引迅速地访问对应于各顶点的信息。在 1.6 节中，我们将讨论一个程序，该程序使用一个符号表在 V 个任意顶点的名字与从 0 到 V-1 的 V 个整数之间建立起一对一映射。有了这个程序，我们就可以将索引用作顶点名(为标记方便)，而不失一般性。有时我们假设顶点集是隐含定义的，即用边集来定义图，并且仅考虑至少在某条边中所包括的顶点。为了避免诸如“有如下边集且包括 10 个顶点的图”这种罗嗦的提法，当顶点个数可以从上下文中很明显地看出来时，我们不会明确地指出顶点个数。在此约定，一个给定图的顶点数总表示为 V，而边数则表示为 E。

我们将图的这一定义作为标准，不过需要说明，在理论上它存在两点简化。首先，在此不允许出现多重边的情况(数学家有时将这种边称为平行边，并将可以包括这种边的图称为多重图)。其次，此定义不允许出现顶点连至顶点自身的边；这种边称为自环。无平行边或自环的图有时称为简单图。

我们在正式定义中使用的是简单图，其原因是这更易于表示其基本属性，而且在许多应用中都不需要平行边和自环。例如，根据给定的顶点数，我们可以确定一个简单图中的边数。

属性 1.1　有 V 个顶点的图至多有 $V(V-1)/2$ 条边。

证明：　在 V^2 组可能的顶点对中，包括 V 个自环，另外每两个不同顶点之间的一条边会重复计两次，因此，边的个数最多为 $(V^2-V)/2=V(V-1)/2$。

如果允许平行边存在，那么这一限制就不再成立：若不是简单图，那么有可能仅包括两个顶点，而连接这两个顶点的边却达到数亿条(或者甚至只有一个顶点，却有数亿个自环)。

对于某些应用，可以考虑将平行边和自环的消除作为实现所必须解决的数据处理问