Robin J. Wilson

Introduction to Graph Theory

5th Edition

图论导论 第5版

PEARSON

世界图书出版公司
www.wpcbj.com.cn

Introduction to Graph Theory

Fifth Edition

Robin J. Wilson

Prentice Hall
is an imprint of

PEARSON

Harlow, England • London • New York • Boston • San Francisco • Toronto
Sydney • Tokyo • Singapore • Hong Kong • Seoul • Taipei • New Delhi
Cape Town • Madrid • Mexico City • Amsterdam • Munich • Paris • Milan

图书在版编目（CIP）数据

图论导论 = Introduction to graph theory：第 5 版：英文/（英）威尔逊（Wilson, R. J.）著．—影印本．—北京：世界图书出版公司北京公司，2014. 8
ISBN 978 - 7 - 5100 - 8451 - 5

Ⅰ．①图…　Ⅱ．①威…　Ⅲ．①图论—英文　Ⅳ．①O157. 5

中国版本图书馆 CIP 数据核字（2014）第 185684 号

书　　名：	Introduction to Graph Theory 5th ed.
作　　者：	Robin J. Wilson
中 译 名：	图论导论 第 5 版
责任编辑：	高蓉　刘慧

出 版 者：	世界图书出版公司北京公司
印 刷 者：	三河市国英印务有限公司
发　　行：	世界图书出版公司北京公司（北京朝内大街 137 号 100010）
联系电话：	010 - 64021602，010 - 64015659
电子信箱：	kjb@ wpcbj. com. cn

开　　本：	24 开
印　　张：	8
版　　次：	2015 年 1 月
版权登记：	图字：01 - 2013 - 5772

书　　号：	978 - 7 - 5100 - 8451 - 5	定　　价：	39. 00 元

前　言

近年来，图论已经成为众多学科中一个重要的数学工具，例如从运筹学和化学到遗传学和语言学，从计算机科学和地理学到社会学和建筑学。同时，图论自身也是数学中的一个重要分支。

鉴于此，人们需要一本能够深入浅出地介绍图论的书，不仅适用于数学家学习图论，而且也能够让非数学专业的学者尽快地了解图论。我希望该最新版能够满足读者这方面的需求。尽管书中稍微困难的习题涉及抽象代数和拓扑，但阅读本书唯一的前提只是需要基础集合论和矩阵理论的基本知识。

本书内容可以分为四个部分。第一部分（1－3 章）是基础知识，包括图和有向图、连通性、欧拉和哈密顿路径和循环以及树的定义和举例。接下来的第二部分（第 4 和第 5 章）介绍平面和着色，重点介绍四色定理。第三部分（第 6 章）主要讲横向理论和连通性，以及它们在网络流中的应用。最后一部分（第 7 章）介绍拟阵，它和前几部分的内容紧密相关，同时也涉及到一些新的进展。

本书重点讲述图论的基础知识。在 300 个习题中，很多是为帮助读者理解内容的常规性练习，剩下的则用来介绍新的结果和思想。读者应该阅读每个习题，无论是否能够完整地求解，因为有些结果在后面会用到。习题号之后有^{S}（上标）的表示给出了该题的答案。每章结尾处列举了一些具有挑战性的难题。

符号▩表示证明结束，粗体铅字表示定义。$|S|$表示集合 S 中元素的个数，Ø 表示空集。

在本版中，很多内容有所改动。重新修订了文字，调整了部分章节。加入了一些新的内容，特别是四色定理和算法，当然也删除了一些。部分改动受益于别人的建议，我要感谢他们。

最后，感谢我以前的学生，要不是他们本书可能会更早完成。感谢 William Shakespeare 等人，他们贴切的和诙谐的名句用在了每章的开始。最重要的，感谢我的妻子 Joy，她为我做了很多事情，尽管那些和图论无关。

R. J. W.

2009 年 11 月

Contents

Go forth, my little book! pursue thy way!
Go forth, and please the gentle and the good.

William Wordsworth

Introduction

The last thing one discovers in writing a book is what to put first.
Blaise Pascal

In this Introduction we provide an intuitive background to the material that we present more formally later on. Terms that appear here in bold-face type are to be thought of as descriptions rather than as definitions; having met them here in an informal setting, you should find them more familiar when you meet them later. So read this Introduction quickly, and then forget all about it!

What is a graph?

We begin by considering Figs 0.1 and 0.2, which depict part of a road map and part of an electrical network.

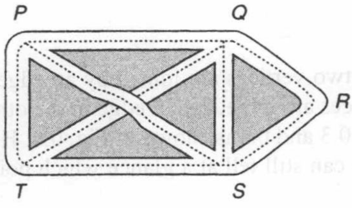

Figure 0.1

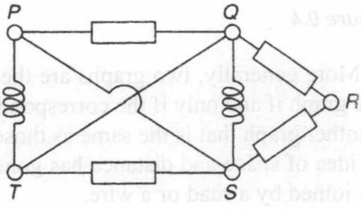

Figure 0.2

Either of these situations can be represented diagrammatically by means of points and lines, as in Fig. 0.3. The points P, Q, R, S and T are called **vertices**, the lines are called **edges**, and the whole diagram is called a **graph**. Note that the intersection of the lines PS and QT is not a vertex, since it does not correspond to a crossroads or to the meeting of two wires. The **degree** of a vertex is the number of edges with that vertex as an end-point; it corresponds in Fig. 0.1 to the number of roads at an intersection. For example, the degree of the vertex P is 3 and the degree of the vertex Q is 4.

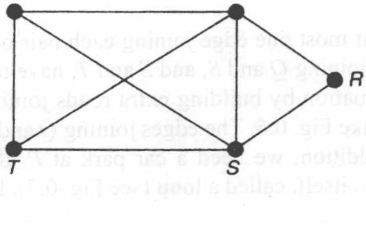

Figure 0.3

The graph in Fig. 0.3 can also represent other situations. For example, if *P*, *Q*, *R*, *S* and *T* represent football teams, then the existence of an edge might correspond to the playing of a game between the teams at its end-points. Thus, in Fig. 0.3, team *P* has played against teams *Q*, *S* and *T*, but not against team *R*. In this representation, the degree of a vertex is the number of games played by that team.

Another way of depicting these situations is to use the graph in Fig. 0.4. Here we have removed the 'crossing' of the lines *PS* and *QT* by redrawing the line *PS* outside the rectangle *PQST*. The resulting graph still tells us whether there is a direct road from one intersection to another, how the electrical network is wired up, and which football teams have played which. The only information we have lost concerns 'metrical' properties, such as the length of a road and the straightness of a wire.

Thus, a graph is a representation of a set of points and of how they are joined up, and any metrical properties are irrelevant. From this point of view, any graphs that represent the same situation, such as those of Figs 0.3 and 0.4, are regarded as the same graph.

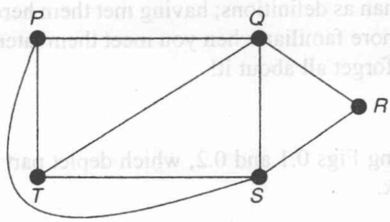

Figure 0.4

More generally, two graphs are the same if two vertices are joined by an edge in one graph if and only if the corresponding vertices are joined by an edge in the other. Another graph that is the same as those in Figs 0.3 and 0.4 is shown in Fig. 0.5. Here all idea of space and distance has gone, but we can still tell at a glance which points are joined by a road or a wire.

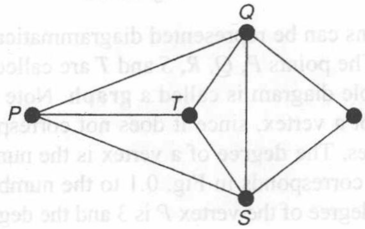

Figure 0.5

In this graph there is at most one edge joining each pair of vertices. Suppose now that in Fig. 0.5 the roads joining *Q* and *S*, and *S* and *T*, have too much traffic to carry. Then we can ease the situation by building extra roads joining these points, and the resulting diagram looks like Fig. 0.6. The edges joining *Q* and *S*, or *S* and *T*, are called **multiple edges**. If, in addition, we need a car park at *P*, then we indicate this by drawing an edge from *P* to itself, called a **loop** (see Fig. 0.7). In this book, graphs may

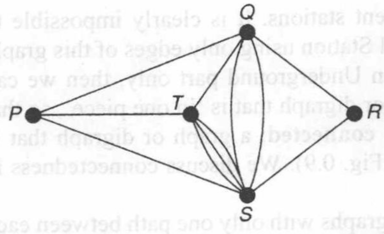

Figure 0.6

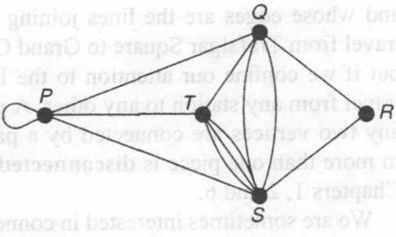

Figure 0.7

contain loops and multiple edges; graphs with no loops or multiple edges, such as the graph in Fig. 0.5, are called **simple graphs**.

The study of **directed graphs** (or **digraphs**, as we abbreviate them) arises from making the roads into one-way streets. An example of a digraph is given in Fig. 0.8, with the directions of the one-way streets indicated by arrows; such a 'directed edge' is called an **arc**. (In this example, there would be chaos at T, but that does not stop us from studying such situations!)

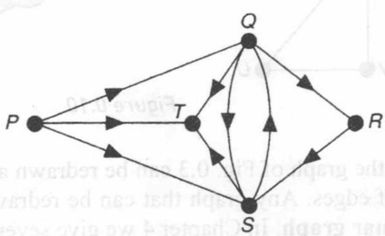

Figure 0.8

Much of graph theory is devoted to 'walks' of various kinds. A **walk** is a 'way of getting from one vertex to another' in a graph or digraph, and consists of a sequence of edges or arcs, one following after another. For example, in Fig. 0.5, $P \to Q \to R$ is a walk of length 2, and $P \to S \to Q \to T \to S \to R$ is a walk of length 5. A walk in which no vertex appears more than once is called a **path**; for example, $P \to T \to S \to R$ is a path. A walk of the form $Q \to S \to T \to Q$, in which no vertex appears more than once, except for the beginning and end vertices which coincide, is called a **cycle**.

In Chapter 2 we also consider walks with some extra property. In particular, we discuss graphs and digraphs containing walks that include every edge exactly once and end back at the initial vertex; such graphs and digraphs are called **Eulerian**. The graph in Fig. 0.5 is not Eulerian, since any walk that includes each edge exactly once (such as $P \to Q \to R \to S \to T \to P \to S \to Q \to T$) must end at a vertex different from the initial one. We also discuss graphs and digraphs containing cycles that pass through every vertex; these are called **Hamiltonian**. For example, the graph in Fig. 0.5 is Hamiltonian; a suitable cycle is $P \to Q \to R \to S \to T \to P$.

Some graphs or digraphs are in two or more parts. For example, consider the graph whose vertices are the stations of the London Underground and the New York Subway, and whose edges are the lines joining adjacent stations. It is clearly impossible to travel from Trafalgar Square to Grand Central Station using only edges of this graph, but if we confine our attention to the London Underground part only, then we can travel from any station to any other. A graph or digraph that is 'in one piece', so that any two vertices are connected by a path, is **connected**; a graph or digraph that is in more than one piece is **disconnected** (see Fig. 0.9). We discuss connectedness in Chapters 1, 2 and 6.

We are sometimes interested in connected graphs with only one path between each pair of vertices. These graphs contain no cycles and are called **trees** (see Fig. 0.10). They generalize the idea of a family tree, and are considered in Chapter 3.

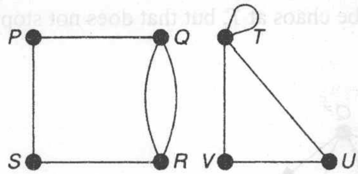

Figure 0.9

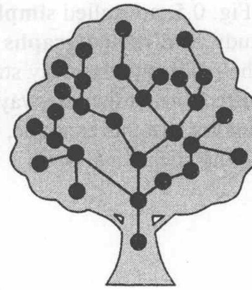

Figure 0.10

Earlier we saw how the graph of Fig. 0.3 can be redrawn as in Figs 0.4 and 0.5 so as to avoid crossings of edges. Any graph that can be redrawn without crossings in this way is called a **planar graph**. In Chapter 4 we give several criteria for planarity. Some of these involve the properties of particular 'subgraphs' of the graph in question; others involve the fundamental notion of duality.

Planar graphs also play an important role in colouring problems. In our 'road-map' graph, let us suppose that Shell, Esso, BP and Gulf wish to erect five garages between them, and that for economic reasons no company wishes to erect two garages at neighbouring corners. Then Shell can build at *P*, Esso can build at *Q*, BP can build at *S*, and Gulf can build at *T*, leaving either Shell or Gulf to build at *R* (see Fig. 0.11). However, if Gulf backs out of the agreement, then the other three companies cannot erect their garages in the specified manner.

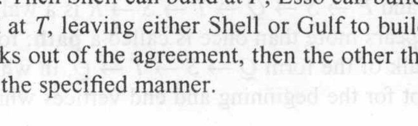

Figure 0.11

We discuss such problems in Chapter 5, where we try to colour the vertices of a simple graph with a given number of colours so that each edge of the graph joins vertices of different colours. If the graph is planar, then we can always colour its vertices in this way with only four colours – this is the celebrated **four-colour theorem**. Another version of this theorem is that we can always colour the countries of any map with four colours so that no two neighbouring countries share the same colour (see Fig. 0.12).

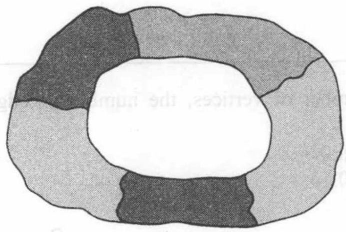

Figure 0.12

In Chapter 6 we investigate the celebrated **marriage problem**, which asks under what conditions a collection of girls, each of whom knows several boys, can be married so that each girl marries a boy she knows. This problem can be expressed in the language of a branch of set theory called 'transversal theory', and is related to problems of finding disjoint paths connecting two given vertices in a graph or digraph.

Chapter 6 concludes with a discussion of network flows and transportation problems. Suppose that we have a transportation network such as in Fig. 0.13, in which P is a factory, R is a market, and the edges of the graph are channels through which goods can be sent. Each channel has a capacity, indicated by a number next to the edge, representing the maximum amount of a commodity that can pass through that channel. The problem is to determine how much of the commodity can be sent from the factory to the market without exceeding the capacity of any channel.

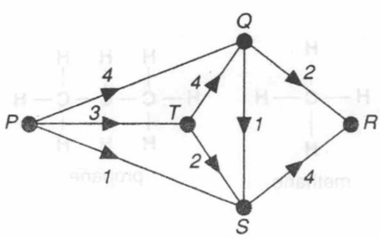

Figure 0.13

We conclude with a chapter on matroids. This ties together the material of the previous chapters, and follows the maxim 'be wise – generalize!' Matroid theory, the study of sets with 'independence structures' defined on them, generalizes both the linear independence of vectors and some results on graphs and transversals from earlier in the book. However, matroid theory is far from being 'generalization for generalization's sake'. On the contrary, it gives us clearer insights into several graphical results involving cycles and planar graphs, and provides simple proofs of results on transversals that are awkward to prove by more traditional methods. Matroids have played an important role in the development of combinatorial ideas in recent years.

We hope that this Introduction has been useful in setting the scene and describing some of the treats that lie ahead. We now embark upon a formal treatment of the subject.

Exercises

0.1[s] Write down the number of vertices, the number of edges and the degree of each vertex in:
(i) the graph in Fig. 0.3;
(ii) the tree in Fig. 0.14.

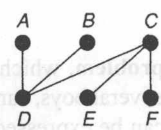

Figure 0.14

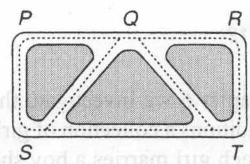

Figure 0.15

0.2 Draw the graph representing the road system in Fig. 0.15, and write down the number of vertices, the number of edges and the degree of each vertex.

0.3[s] Figure 0.16 represents the chemical molecules of methane (CH_4) and propane (C_3H_8).
(i) Regarding these diagrams as trees, what can you say about the vertices representing carbon atoms (**C**) and hydrogen atoms (**H**)?
(ii) Draw a tree that represents the molecule with formula C_2H_6 (hexane).
(iii) There are two different chemical molecules with formula C_4H_{10}. Draw trees that represent these molecules.

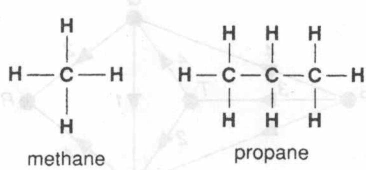

Figure 0.16

0.4 Draw a graph that represents the family tree in Fig. 0.17.

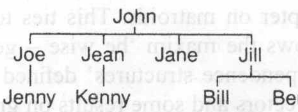

Figure 0.17

0.5[s] John likes Joan, Jean and Jane; Joe likes Jane and Joan; Jean and Joan like each other. Draw a digraph illustrating these relationships between John, Joan, Jean, Jane and Joe.

0.6 Snakes eat frogs and birds eat spiders; birds and spiders both eat insects; frogs eat snails, spiders and insects. Draw a digraph representing this predatory behaviour.

0.7 Draw a graph with vertices $A, B, \ldots, M$ that shows the various routes that one can take when tracing the Hampton Court maze in Fig. 0.18.

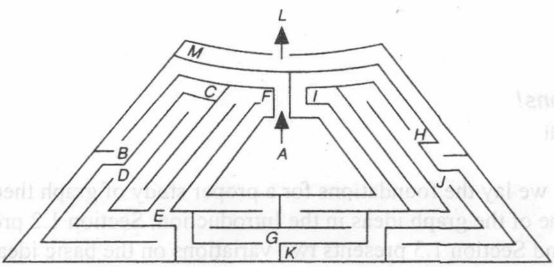

Figure 0.18

Chapter 1

Definitions and examples

I hate definitions!

Benjamin Disraeli

In this chapter, we lay the foundations for a proper study of graph theory. Section 1.1 formalizes some of the graph ideas in the Introduction, Section 1.2 provides a variety of examples, and Section 1.3 presents two variations on the basic idea. In Section 1.4 we show how graphs can be used to represent and solve three problems from recreational mathematics. More substantial applications are deferred until Chapters 2 and 3, when we have more machinery at our disposal.

1.1 Definitions

A **simple graph** G consists of a non-empty finite set $V(G)$ of elements called **vertices** (or **nodes** or **points**) and a finite set $E(G)$ of distinct unordered pairs of distinct elements of $V(G)$ called **edges** (or **lines**). We call $V(G)$ the **vertex-set** and $E(G)$ the **edge-set** of G. An edge $\{v, w\}$ is said to **join** the vertices v and w, and is usually abbreviated to vw. For example, Fig. 1.1 represents the simple graph G whose vertex-set $V(G)$ is $\{u, v, w, z\}$, and whose edge-set $E(G)$ consists of the edges uv, uw, vw and wz.

In any simple graph there is at most one edge joining a given pair of vertices. However, many results for simple graphs also hold for more general objects in which two vertices may have several edges joining them; such edges are called **multiple edges**. In addition, we may remove the restriction that an edge must join two *distinct* vertices, and allow **loops** – edges joining a vertex to itself. The resulting object, with loops and multiple edges allowed, is called a **general graph** – or, simply, a **graph** (see Fig. 1.2). Note that every simple graph is a graph, but not every graph is a simple graph.

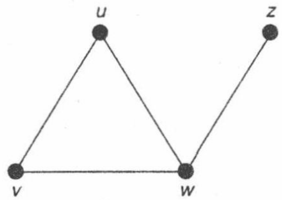

Figure 1.1

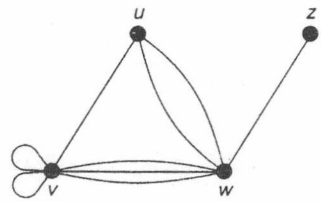

Figure 1.2

Thus, a **graph** G consists of a non-empty finite set $V(G)$ of elements called **vertices** and a finite family $E(G)$ of unordered pairs of (not necessarily distinct) elements of $V(G)$ called **edges**; the use of the word 'family' permits the existence of multiple edges. We call $V(G)$ the **vertex-set** and $E(G)$ the **edge-family** of G. An edge $\{v, w\}$ is said to **join** the vertices v and w, and is again abbreviated to vw. Thus in Fig. 1.2, $V(G)$ is the set $\{u, v, w, z\}$ and $E(G)$ consists of the edges uv, vv (twice), vw (three times), uw (twice) and wz. Note that each loop vv joins the vertex v to itself. Although we sometimes need to restrict our attention to simple graphs, we shall prove our results for general graphs whenever possible.

Remark. The language of graph theory is not standard – all authors have their own terminology. Some use the term 'graph' for what we call a simple graph, while others use it for graphs with directed edges, or for graphs with infinitely many vertices or edges; we discuss these variations in Section 1.3. Any such definition is perfectly valid, provided that it is used consistently. In this book:

All graphs are finite and undirected, with loops and multiple edges allowed unless specifically excluded.

Isomorphism

Two graphs G_1 and G_2 are **isomorphic** if there is a one–one correspondence between the vertices of G_1 and those of G_2 such that the number of edges joining any two vertices of G_1 equals the number of edges joining the corresponding vertices of G_2. For example, the two graphs in Fig. 1.3 are isomorphic, under the correspondence

$$u \leftrightarrow l,\ v \leftrightarrow m,\ w \leftrightarrow n,\ x \leftrightarrow p,\ y \leftrightarrow q,\ z \leftrightarrow r.$$

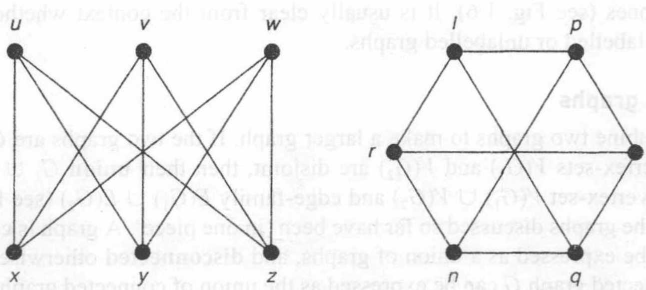

Figure 1.3

For many problems, the labels on the vertices are unnecessary and we drop them. We then say that two 'unlabelled graphs' are isomorphic if we can assign labels to their vertices so that the resulting 'labelled graphs' are isomorphic. For example, we regard the unlabelled graphs in Fig. 1.4 as isomorphic, since the labelled graphs in Fig. 1.3 are isomorphic.

The difference between labelled and unlabelled graphs becomes more apparent when we try to count them. For example, if we restrict ourselves to graphs with three vertices, then there are eight different labelled graphs (see Fig. 1.5), but only four

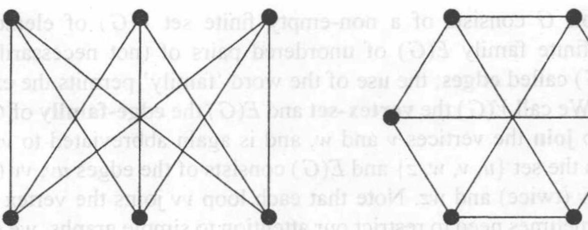

Figure 1.4

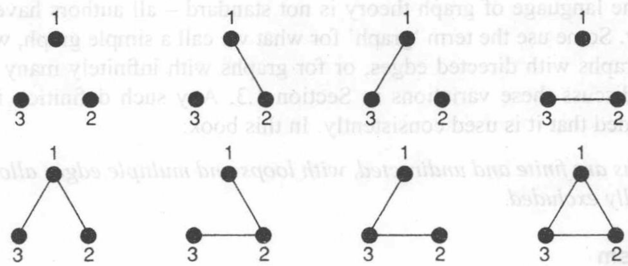

Figure 1.5

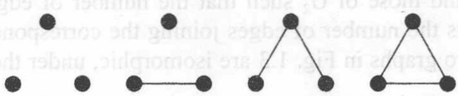

Figure 1.6

unlabelled ones (see Fig. 1.6). It is usually clear from the context whether we are referring to labelled or unlabelled graphs.

Connected graphs

We can combine two graphs to make a larger graph. If the two graphs are G_1 and G_2 and their vertex-sets $V(G_1)$ and $V(G_2)$ are disjoint, then their **union** $G_1 \cup G_2$ is the graph with vertex-set $V(G_1) \cup V(G_2)$ and edge-family $E(G_1) \cup E(G_2)$ (see Fig. 1.7).

Most of the graphs discussed so far have been 'in one piece'. A graph is **connected** if it cannot be expressed as a union of graphs, and **disconnected** otherwise. Clearly, any disconnected graph G can be expressed as the union of connected graphs, each of which is called a **component** of G; a disconnected graph with three components is shown in Fig. 1.8.

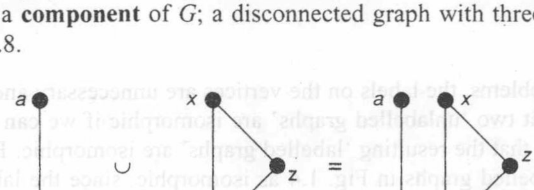

Figure 1.7

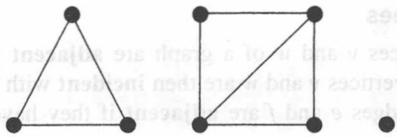

Figure 1.8

When proving results about graphs in general, we can often obtain the corresponding results for connected graphs and then apply them to each component separately. A table of all the unlabelled connected simple graphs with up to five vertices is given in Fig. 1.9.

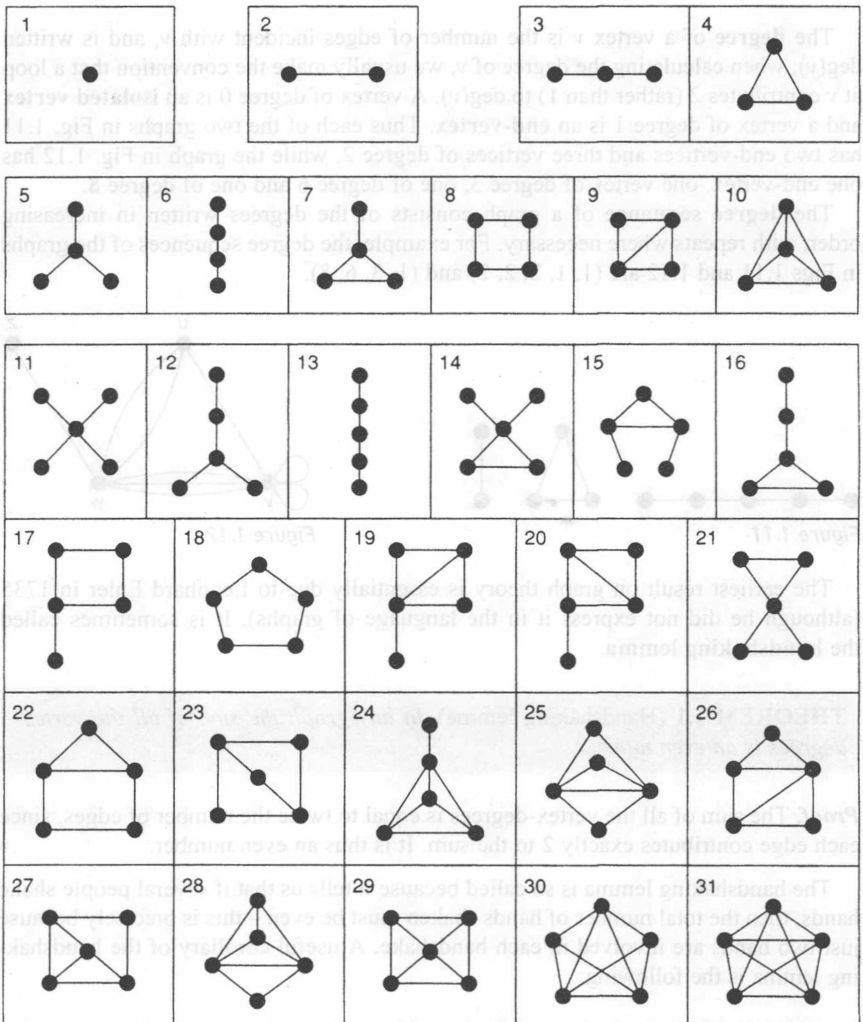

Figure 1.9