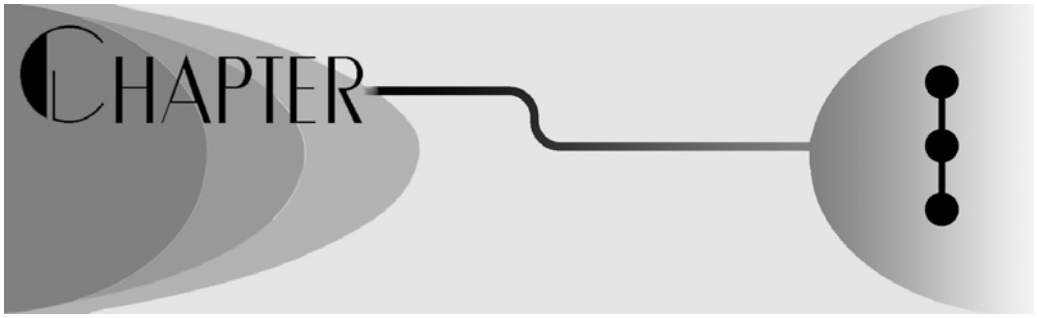
Contents



Introduction to Graph Theory

Chapter Goals

Motivate the subject of graph theory.	Provide an intuitive feel for graph theory.
Present numerous applications of graph theory.	Describe basic mathematical preliminaries.
Give a variety of examples of graphs.	Prove the Hand Shaking Theorem and related results.
Introduce a number of the fundamental definitions in graph theory.	Present a complete example illustrating how graph theory can be used to solve a specific problem.

1.1 Introduction

At the beginning of each chapter, we provide a road map of the chapter's contents. We do this with a description of the contents of the various sections of the chapter.

In §1.2 we provide several examples to motivate the study of graph theory. A number of important mathematical preliminiaries are covered in §1.3. We

define graphs in §1.4. In §1.5 a number of important examples are presented. Next we examine some properties of graphs. In §1.6 we look at the concept of degree. We also study regular graphs in this section. Various portions of graphs are explored in §1.7. This is followed in §1.8 with an explanation of how graph theory can be used to model various problems. One of the key features of graphs is that they easily model many different types of problems. Graph theory has a long, interesting history. §1.9 provides a brief glimpse of this history. Finally, we examine the contents of the remainder of the book in §1.10.

1.2 Why Study Graphs?

To motivate the study of graphs, first we look at several problems that can easily be modeled using *graph theory*. Technical details and definitions for the intuitive concepts described in this section are provided in §1.4. With its inherent simplicity, graph theory has a wide range of applications in numerous areas including biological science, computer science, economics, engineering, linguistics, mathematics, medicine, physical science, and social science. A graph can be used to represent almost any physical situation involving relationships among discrete objects. It can also be used to model many abstract situations involving relationships among objects. The following are representative problems chosen from the hundreds of such applications of graph theory.

The Königsberg bridge is perhaps the best-known example in graph theory. It was a long-standing open problem until solved by Leonhard Euler in 1736 by means of a graph. Euler wrote the very first paper about graphs and thus became the originator of graph theory.

Problem 1.1
Königsberg Bridge

The setting for the Königsberg bridge problem is depicted in Figure 1.1. Two islands, C and D, formed by the Pregel River in Königsberg[1] *were connected to each other and to the banks A and B with seven bridges. This is shown in Figure 1.1. The problem was to start at any of the four land areas of the city, A, B, C, or D, walk over each of the seven bridges* exactly *once, and return to the starting point. (No swimming was allowed.)*

[1] This former capital of East Prussia is now called Kaliningrad and lies in West Russia.

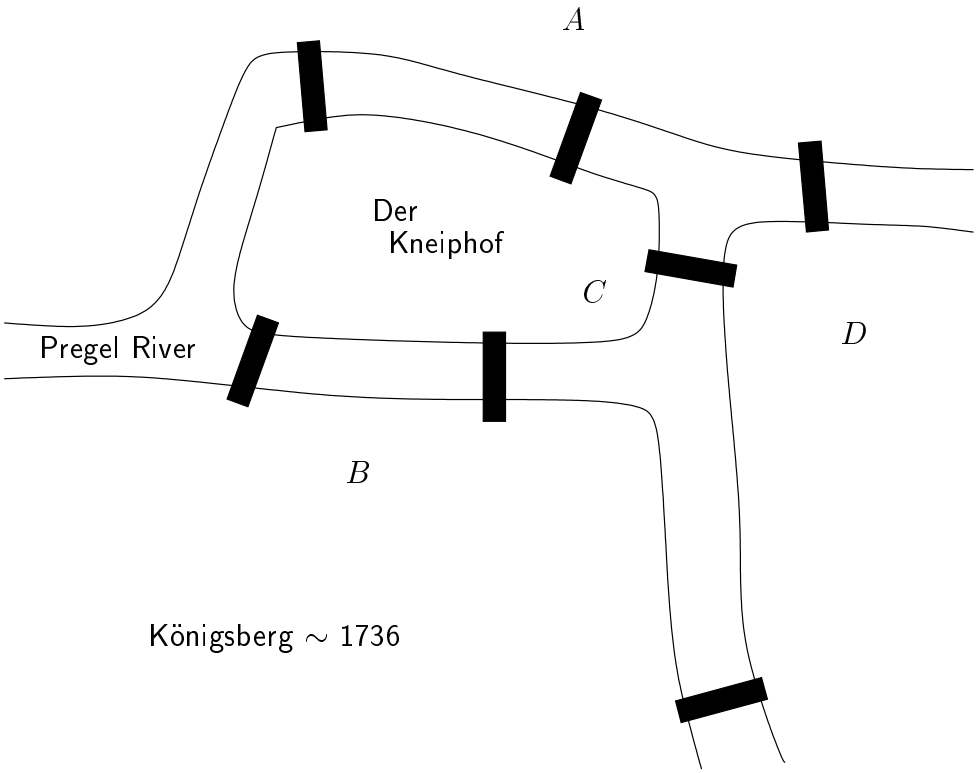


Figure 1.1: Königsberg bridge problem.

Euler represented this situation with a graph as shown in Figure 1.2. The vertices (black circles) represent the land areas and the edges (solid lines) represent the bridges.

Note that the Königsberg bridge problem is the same as the problem of drawing the lines in the figure without lifting the pen from the paper and without retracing a line. We all have been confronted with similar problems at one time or another. As we shall soon see in Chapter 2, Euler proved that a solution for this problem does not exist.

The next application deals with discovering communities on the World Wide Web.

Problem 1.2

World Wide Web Communities

The World Wide Web can be modeled as a graph. In one such model, Web

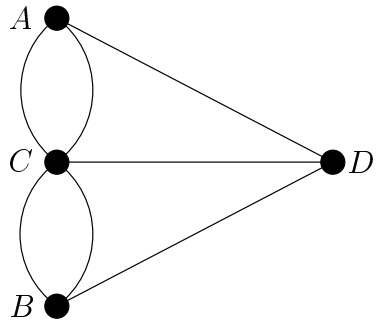


Figure 1.2: Graph of Königsberg bridge problem.

pages can be represented by vertices and hyperlinks between them by edges in the graph. By examining certain properties of this "Web graph," we can discover some interesting information. For example, the structure shown in Figure 1.3 is termed a Web community*. In graph theory this is known as a* complete bipartite graph*.*

On the lefthand side of the figure, five vertices are displayed with labels travelagencyX.html*, where* $1 \leq X \leq 5$*. Each label travelagencyX.html represents an HTML file for a given travel agency. On the righthand side of the figure, three vertices are displayed with labels* www.continental.com*,* www.delta.com*, and* www.united.com*. These represent the Web sites of several well-known airlines. Notice that each HTML file has an edge between it and every airline. No two travel agencies have edges between them, and no two airlines have edges between them. This graph structure is common on the Web between competitors in the same industry. Companies or rivals usually do not have hyperlinks on their Web pages to their competition. For example, Delta has no hyperlink to United Airlines. Travel agents may have hyperlinks from their Web pages to all of the airlines as shown in Figure 1.3. Notice none of the travel agents are linked together.*

This example illustrates how graph theory can assist us in discovering Web communities. We can uncover such communities by finding complete bipartite subgraphs in the Web graph. Web community information could be used for marketing purposes or for examining the relationships among companies in a given industry. Other facets of the World Wide Web can also be modeled using graph theory.

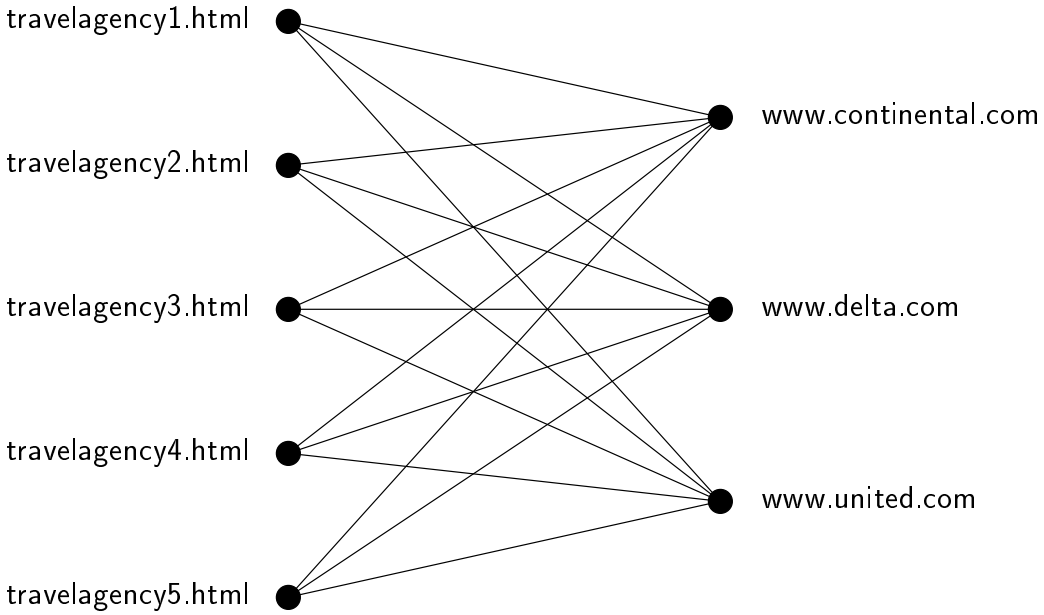


Figure 1.3: World Wide Web community.

The next application we present deals with connecting electrical utilities.

Problem 1.3
Electrical Utilities Connection

There are three houses depicted in Figure 1.4. They are denoted H_1, H_2, and H_3. Each house is to be connected to each of three utilities: water (W), gas (G), and electricity (E) by means of conduits.

Is it possible to make such connections without any edge-crossings of the conduits?

Figure 1.5 shows how this problem can be represented by a graph—the conduits are shown as edges while the houses and utility supply centers are vertices. As we shall see in Chapter 7, the graph in Figure 1.5 cannot be drawn in the plane without edges crossing each other. Thus the answer to the problem is no. More complex real life problems are modeled by generalizing this example.

This next application deals with job assignments.

Problem 1.4
Job Assignments

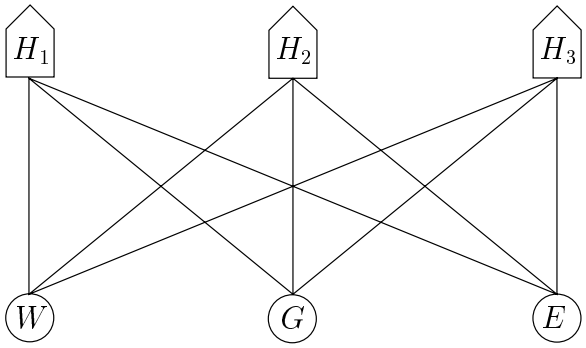


Figure 1.4: Three-utilities problem.

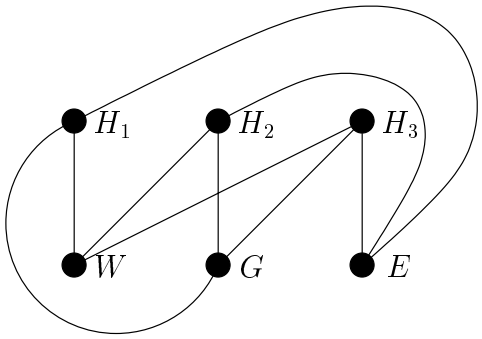


Figure 1.5: Graph of three-utilities problem.

A company wants to hire people for five different jobs. We denote the jobs by J_i for $1 \leq i \leq 5$. There are seven applicants denoted by A_j for $1 \leq j \leq 7$. In the bipartite graph *in Figure 1.6, the* vertices *in the upper group, labeled $J_1, \ldots, J_5$, represent the different jobs, and the vertices in the lower group, labeled $A_1, \ldots, A_7$, represent the seven different applicants. An* edge *between J_i and A_j means that applicant A_j is qualified to perform job J_i.*

Is the company able to hire five qualified applicants from these seven, and thereby fill their open positions as desired?

In Chapter 11 we will derive necessary and sufficient conditions for such job-assignment-problems to have a positive solution. Clearly, solving practical problems in hiring is important. This discussion illustrates that graph theory is a good approach to solving such problems.

Next we use a graph to determine the minimum nuber of warehouses needed

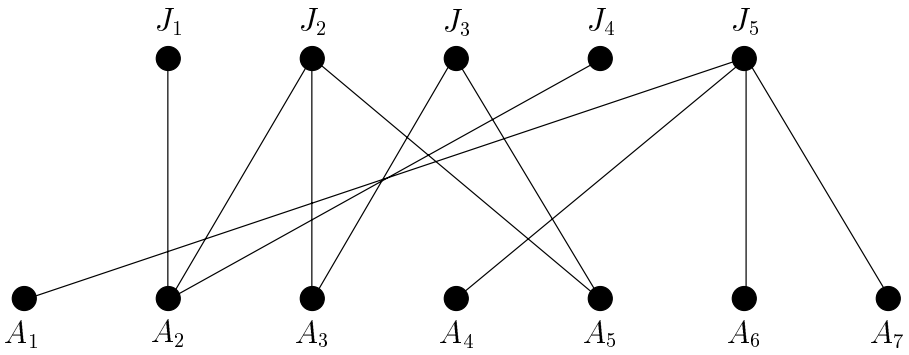


Figure 1.6: Bipartite graph describing job qualification.

to store volatile chemicals.

Problem 1.5

Storing Volatile Chemicals

A chemical factory produces chemicals $C_1, C_2, \ldots, C_7$. For security reasons some of the chemicals should not be stored in the same warehouse. The graph in Figure 1.7 represents each chemical as a vertex. An edge between the chemicals C_i and C_j indicates a grave danger of storing these chemicals in the same warehouse.

What is the minimum number of warehouses the factory needs in order to safely store its chemical products?

By thinking of each warehouse as a color *assigned to each vertex, we can reformulate the problem. The problem boils down to using the least number of colors to color the vertices of the graph such that no two adjacent vertices receive the same color. The least number of colors needed to color a graph in this manner is called the* chromatic number *of the graph.*

The graph shown in Figure 1.7 can be colored using four colors by coloring vertex C_1 with color 1, C_2 with color 2, C_3 with color 3, C_4 with color 1, C_5 with color 2, C_6 with color 3, and C_7 with color 4. Note that the graph cannot be colored using three colors.

Colorings are discussed in Chapter 11. There we develop a systematic way of finding the chromatic number for any given graph, and study graph colorings in various other settings. In this application the chromatic number tells us the minimum number of warehouses required, and an actual coloring tells us which

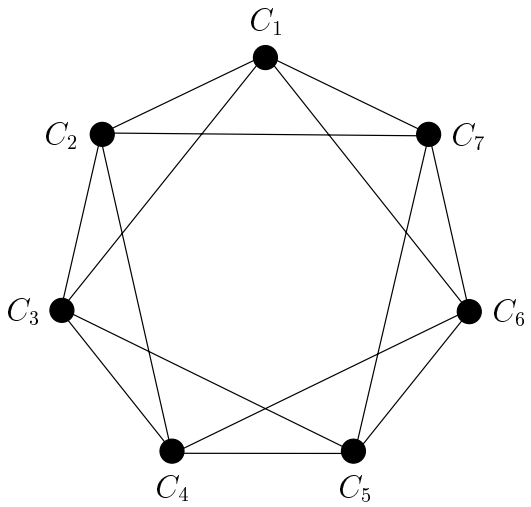


Figure 1.7: Connected chemicals should not be stored together.

chemicals to store in each warehouse. Since warehouses are expensive to build, it is clear that finding the fewest warehouses to construct is extremely important. It is straightforward to generalize this example to more complex problems.

The next application uses a graph to model seating arrangements.

Problem 1.6

Seating Arrangements

Nine members of a new club meet each day for lunch at a round table. They sit so that every member has different neighbors at each lunch. How many days can such arrangements be found?

This situation can be represented by a graph with nine vertices. Each vertex represents a member, and an edge joining two vertices represents the relationship of sitting next to each other. Figure 1.8 shows two possible seating arrangements:

- *1 2 3 4 5 6 7 8 9 1 (solid lines)*
- *1 3 5 2 7 4 9 6 8 1 (dashed lines)*

It can be shown by graph-theoretic considerations that there are only two more arrangements possible. They are

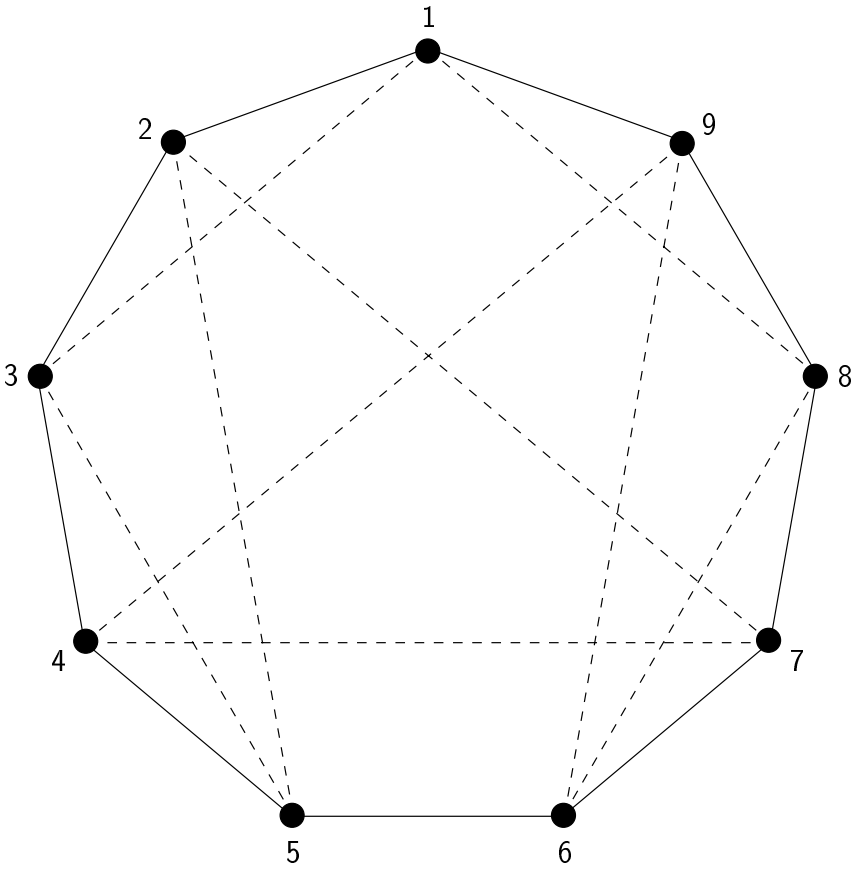


Figure 1.8: Arrangements at a dinner table.

▷ *1 5 7 3 9 2 8 4 6 1*

▷ *1 7 9 5 8 3 6 2 4 1*

In Chapter 3 we will show that for such a club with n members, the maximum number of such seating arrangements possible is $\lfloor (n-1)/2 \rfloor$. (Here $\lfloor x \rfloor$ denotes the largest integer that is less than or equal to x for any real number x.)

In this section we have presented a handful of simple, practical applications of graph theory. You can probably imagine many other problems that can be modeled using these techniques. Although the basic concepts of graph theory are readily applied even by the nonspecialist, it is necessary to formalize the

notions highlighted by these examples to tackle more challenging and interesting problems.

1.3 Mathematical Preliminaries

In this section we present a number of preliminaries. We give several definitions and provide examples to illustrate how these definitions come about. Throughout the book we describe various concepts followed by examples illustrating the ideas, and then we include interesting relationships among the concepts. This approach will make it easy for you to learn the essentials of graph theory. First though, we need to specify some technical set theoretic conventions, which every undergraduate majoring in computer science, engineering, mathematics, or related fields should learn.

A *set* is a collection of objects. An object in a set is called an *element*. The set containing no elements is called the *empty set* and is denoted by $\emptyset$. Let A and B be sets. If a is an element contained in a set A, we denote this by $a \in A$. If for every $a \in A$, it is also true that $a \in B$, we say that A is a *subset* of B and denote this by $A \subseteq B$.

We have the following set operations: The *union* of A and B, denoted by $A \cup B$, is the set of those elements that are either in A or in B. The *intersection* of A and B, denoted by $A \cap B$, is the set consisting of those elements that are contained in both A and B. If $A \cap B$ is empty, then we say that A and B are *disjoint*. The *difference* of A and B, denoted by $A \setminus B$, consists of those elements that are contained in A and not in B. The *symmetric difference* of A and B, denoted by $A \vartriangle B$, is equal to $(A \setminus B) \cup (B \setminus A)$. It consists of those elements that are contained in exactly one of the sets A and B. It makes sense to generalize the definitions of union, intersection, and symmetric difference to a collection of finitely many sets rather than just two. That is, one can show that if $A_1, A_2, \ldots, A_n$ is a finite collection of sets then the notions of $A_1 \cup \cdots \cup A_n$, $A_1 \cap \cdots \cap A_n$, and $A_1 \vartriangle \cdots \vartriangle A_n$ all are sensible. The last one denotes the set of elements that are contained in an odd number of the sets $A_1, \ldots, A_n$. (Can you see why?)

A set S containing finitely many elements is a *finite set*. We denote the number of elements in the set S by $|S|$. For example, the set $\{1, 2, 3, 4, 5\}$ has five elements. That is, $|\{1, 2, 3, 4, 5\}|$ equals five. If S is infinite, that is, not

finite, then $|S| = \infty$. The set of natural numbers, $\mathbb{N} = \{1, 2, 3, \ldots\}$, is an infinite set. Nearly all sets in this book will be finite sets. If S is a set, then the set of all subsets of S, denoted by $\mathbf{P}(S)$, is called the *power set* of S. For example, if $S = \{1, 2, 3\}$, then

$$\mathbf{P}(S) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}.$$

Let $n \in \mathbb{N}$ and $a_1, a_2, \ldots, a_n$ be any n objects. Then $(a_1, a_2, \ldots, a_n)$ is an *ordered n-tuple* or just *n-tuple* for short. For each $i \in \{1, 2, \ldots, n\}$, we say a_i is the i-th *component* of $(a_1, a_2, \ldots, a_n)$.

There are several other sets of numbers that we use in this book. The set of integers is $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$. We use $\mathbb{R}$ to denote the real numbers. The notation $[a; b]$ denotes the of real numbers x such that $a \leq x \leq b$. The notation $[a; b[$ denotes $[a; b] \setminus \{b\}$, $]a; b]$ denotes $[a; b] \setminus \{a\}$, and $]a; b[$ denotes $[a; b] \setminus \{a, b\}$. The set of rational numbers is

$$\mathbb{Q} = \{a/b : a, b \in \mathbb{Z} \text{ and } b \neq 0\}.$$

The set of complex numbers is

$$\mathbb{C} = \{x + yi : x, y \in \mathbb{R}\},$$

where $i = \sqrt{-1}$ denotes the imaginary number satisfying $i^2 = -1$.

A *map* (also called *function* or *mapping*) $f : X \mapsto Y$ is *injective* or *one-to-one* if $f(x_1) = f(x_2)$ always implies that $x_1 = x_2$. Also, f is *surjective* or *onto* if for every $y \in Y$, there is an $x \in X$ such that $y = f(x)$. A map that is both injective and surjective is called *bijective*. Such functions are also called *bijections* or *one-to-one correspondences*.

☞ **Example 1.7**
Consider the function $f : \mathbb{N} \mapsto \mathbb{N}$ defined by $f(n) = n^2$. Suppose $f(n) = f(m)$. This means $n^2 = m^2$. Since $n, m \in \mathbb{N}$, this implies $n = m$. Therefore, f is an injective mapping. Notice f is not onto $\mathbb{N}$, since there is no $n \in \mathbb{N}$ such that $n^2 = 3$. This means f is not a one-to-one correspondence. It is worth noting that if f were defined on the integers, it would not be one-to-one.

The map from X to X that has a value of x for each $x \in X$ is called the *identity map*. The identity map on X is denoted by $\mathbf{I}_X : X \mapsto X$. $\mathbf{I}_X$ is a bijective map for any X. Let $f : X \mapsto Y$ and $g : Y \mapsto Z$ be any two maps.

The *composition* of g with f denoted by $g \circ f : X \mapsto Z$ is the mapping such that $(g \circ f)(x) = g(f(x))$ for each $x \in X$. If $f : X \mapsto Y$ is a bijective map, then there is an unique map, denoted by $f^{-1} : Y \mapsto X$, such that $f^{-1} \circ f = \mathbf{I}_X$ and $f \circ f^{-1} = \mathbf{I}_Y$. The map f^{-1} is called the *inverse* of the map f. Finally, if $f : X \to Y$ is a map and $X' \subseteq X$, then the *restriction* of f to X' is the map $f_{X'} : X' \to Y$ defined by $f_{X'}(x) = f(x)$ for each $x \in X'$.

☞ **Example 1.8**
Let $f : \mathbb{N} \mapsto \mathbb{R}$ be defined by $f(n) = n$. Let $g : \mathbb{R} \mapsto \mathbb{R}$ be defined by $g(x) = -x + 1$. Then $g \circ f : \mathbb{N} \mapsto \mathbb{R}$ is defined. In fact,

$$(g \circ f)(n) = g(f(n)) = g(n) = -n + 1.$$

Note that $f \circ g$ is not defined. It is easy to see that g is a bijection of $\mathbb{R}$. So, the inverse of g exists. In fact, $g^{-1}(x) = -x + 1$. Notice

$$(g \circ g^{-1})(x) = g(g^{-1}(x)) = g(-x + 1) = -(-x + 1) + 1 = x - 1 + 1 = x.$$

We see $g \circ g^{-1} = g^{-1} \circ g = \mathbf{I}_{\mathbb{R}}$. The restriction of g to $\mathbb{N}$ is defined by $g_{\mathbb{N}}(x) = g(x)$ for each $x \in \mathbb{N}$.

Let n be a natural number. The notation $n!$ denotes the product

$$1 \cdot 2 \cdots (n-1) \cdot n.$$

Note $0!$ is defined to be 1. The notation $n!$ is read *n-factorial*. Let n and k be natural numbers with k less than or equal to n. The notation

$$\binom{n}{k}$$

denotes the number of ways to choose a k-element subset from a set containing n elements. This value is equal to

$$\frac{n!}{k!(n-k)!}.$$

(Can you see why?) The notation is read n *choose* k. From this we can see that

$$\binom{n}{0} = \binom{n}{n} = 1,$$

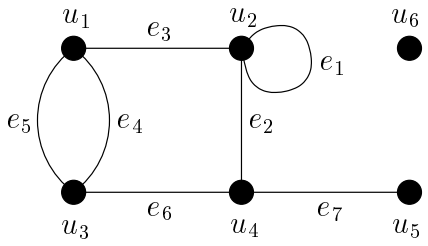


Figure 1.9: A graph with six vertices and seven edges.

$$\binom{n}{k} = \binom{n}{n-k}, \text{ and}$$

$$\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1}.$$

We now have the foundations to begin looking at graphs more closely.

1.4 What is a Graph?

Having settled on a few technical preliminaries, we can start to talk about graphs. What is a graph? Informally, a graph is simply an ordered pair (V, E) consisting of two sets. Here V is the set of *vertices* or *nodes*, and E is the set of *edges* connecting the vertices together in some way. In §1.2 we saw a number of pictorial representations of graphs. To be precise, the following definition captures the essentials of the intuitive description of graphs.

Definition 1.9
A graph *or a* general graph *is an ordered triple* $G = (V, E, \phi)$*, where*

1. $V \neq \emptyset$*,*
2. $V \cap E = \emptyset$*, and*
3. $\phi : E \mapsto \mathbf{P}(V)$ *is a map such that* $|\phi(e)| \in \{1, 2\}$ *for each* $e \in E$*. This map is called an* edgemap*.*

The elements of V *are the* vertices *of* G*, and the elements of* E *are the* edges *of* G*. The vertices in* $\phi(e)$ *are called the* endvertices *of* e*.*

We look at an example to illustrate this definition.

☞ **Example 1.10**
The most common representation of a graph is a diagram. In such a diagram the vertices are represented as points and each edge as a line segment or curve joining its endvertices. Often, this diagram itself is referred to as the graph. Consider the graph represented by the diagram in Figure 1.9. In this case we have $G = (V, E, \phi)$*, where* $V = \{u_1, u_2, \ldots, u_6\}$*,* $E = \{e_1, e_2, \ldots, e_7\}$*, and where the edgemap* ϕ *is as follows:*

$$\begin{aligned}
\phi(e_1) &= \{u_2\} \\
\phi(e_2) &= \{u_2, u_4\} \\
\phi(e_3) &= \{u_1, u_2\} \\
\phi(e_4) &= \phi(e_5) = \{u_1, u_3\} \\
\phi(e_6) &= \{u_3, u_4\} \\
\phi(e_7) &= \{u_4, u_5\}
\end{aligned}$$

In the next definition we present some common graph-theoretic terminology so that we can discuss the various aspects of graphs.

Definition 1.11
Two vertices u *and* v *in* V *are* adjacent *or* neighbors*, if they are the endvertices of some edge* $e \in E$*. That is, they are adjacent if there is an* $e \in E$ *such that* $\phi(e) = \{u, v\}$*.*

Two edges e *and* f *are* incident *with each other, if they have a common endvertex. That is,* $\phi(e) \cap \phi(f)$ *is nonempty.*

A loop *is an edge* e *whose endvertices are equal. That is,* $|\phi(e)|$ *equals one.*

We say that $E' \subseteq E$ *is a set of* multiple edges *or* parallel edges *if* $|E'|$ *is greater than or equal to two and all* $e' \in E'$ *have the same set of endvertices. That is,* $\phi(e')$ *equals* $\phi(f')$ *for all* $e', f' \in E'$*.*

A vertex u *is called* isolated *if it is not an endvertex of any edge. That is,* $u \notin \phi(e)$ *for all* $e \in E$*.*

We revisit Figure 1.9 to see examples of these concepts.

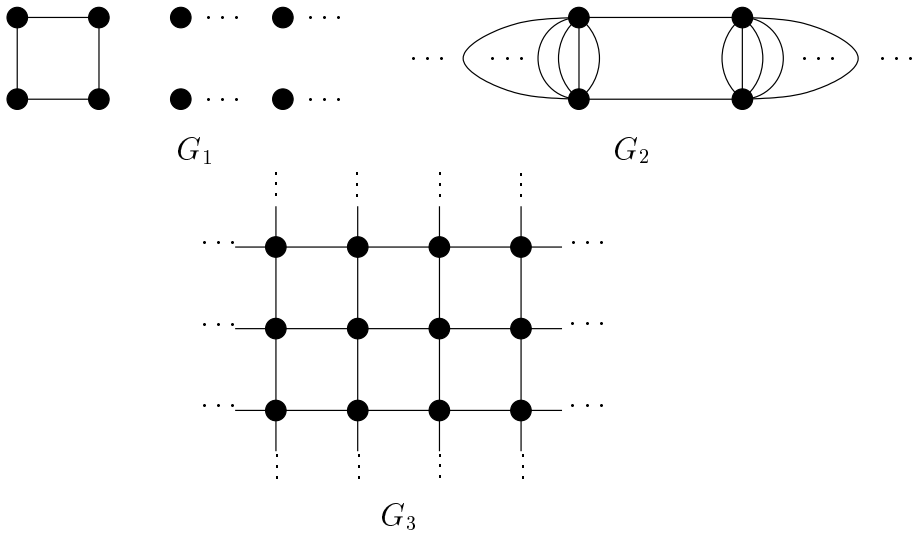


Figure 1.10: Portions of three infinite graphs.

☞ **Example 1.12**
In Figure 1.9 the endvertices of e_2 are u_2 and u_4. The vertices u_1 and u_3 are adjacent, but u_1 and u_4 are not. So, u_1 and u_3 are neighbors, but u_1 and u_4 are not. The vertex u_2 is adjacent to itself. The edges e_2 and e_3 are incident with each other, but e_2 and e_4 are not. The edge e_1 is incident both with itself and e_2. The edge e_1 is the only loop in this graph. The set $\{e_4, e_5\}$ consists of two parallel edges. The vertex u_6 is isolated.

We next state an important convention.

✎ **Convention 1.13**
For a graph G, the set $V(G)$ will always denote the set of vertices of G, and $E(G)$ will always denote the set of edges of G.

Note that in Example 1.10, both $V(G)$ and $E(G)$ are finite sets. However, in our definition of a graph, neither the vertex set $V(G)$ nor the edge set $E(G)$ need to be finite. In most applications these sets are finite. A graph with a finite number of vertices and edges is called a *finite graph*; otherwise, it is called an *infinite graph*. Portions of three infinite graphs are shown in Figure 1.10.

➤ **Note 1.14**
In this book we shall focus on the study of finite graphs. Unless otherwise stated,

Figure 1.11: Null graph N_n.

the term "graph" will always mean finite graph.

Many problems in graph theory can be reduced to the study of those graphs with no loops nor multiple edges. These graphs form an important class among graphs called *simple graphs*. Simple graphs can be defined without the edgemap ϕ in Definition 1.9 since each edge is in a one-to-one correspondence with its pair of endvertices, which are always distinct. Because simple graphs are important and used frequently, we state their formal definition separately.

Definition 1.15
A simple graph *is an ordered pair $G = (V, E)$, where V is a nonempty set of vertices and E is a set of unordered pairs of distinct vertices of V. That is,*

$$E \subseteq \{X : X \subseteq V, |X| = 2\} = \{\{u, v\} : u, v \in V, u \neq v\}.$$

The notation above is read "the set of X such that X is a subset of V and X has two elements." The second part of this line is read "the set containing two element sets having elements u and v such that both u and v are in the set V, and u does not equal v." A non-literal reading of the last part is "the set containing sets having two distinct elements both of which are in the set V."

1.5 Important Examples of Graphs

We give several examples of common simple graphs below. In the examples that follow $m, n \in \{1, 2, \ldots\}$.

☞ **Example 1.16**
The null graph *on n vertices is the simple graph $N_n = (V, E)$, with the sets $V = \{u_1, u_2, \ldots, u_n\}$ and $E = \emptyset$.*

☞ **Example 1.17**
The path graph *on n vertices is the simple graph $P_n = (V, E)$, where $V = \{u_1, u_2, \ldots, u_n\}$ and $E = \{\{u_1, u_2\}, \{u_2, u_3\}, \ldots, \{u_{n-1}, u_n\}\}$.*

☞ **Example 1.18**
In this case assume that n is a natural number greater than or equal to two. The cycle *on n vertices is the simple graph $C_n = (V, E)$, where*

$$V = \{u_1, \ldots, u_n\}$$

and

$$E = \{\{u_1, u_2\}, \{u_2, u_3\}, \ldots, \{u_{n-1}, u_n\}, \{u_n, u_1\}\}.$$

Note that C_n has one more edge than P_n.

☞ **Example 1.19**
The complete graph *on n vertices is the simple graph $K_n = (V, E)$, where $V = \{u_1, u_2, \ldots, u_n\}$ and every pair of distinct vertices are connected by an edge. That is, E equals $\{\{u_i, u_j\} : i, j \in \{1, 2, \ldots, n\}, i < j\}$.*

☞ **Example 1.20**
The complete bipartite (m, n)-graph *on $m+n$ vertices is the simple graph $K_{m,n} = (V, E)$, where $V = \{u_1, u_2, \ldots, u_m\} \cup \{v_1, v_2, \ldots, v_n\}$ and $E = \{\{u_i, v_j\} : i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}\}$.*

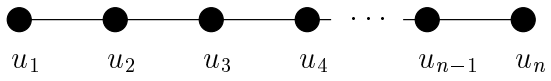


Figure 1.12: Path graph P_n.

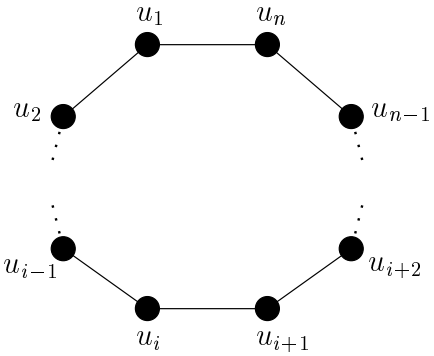


Figure 1.13: Cycle graph C_n.

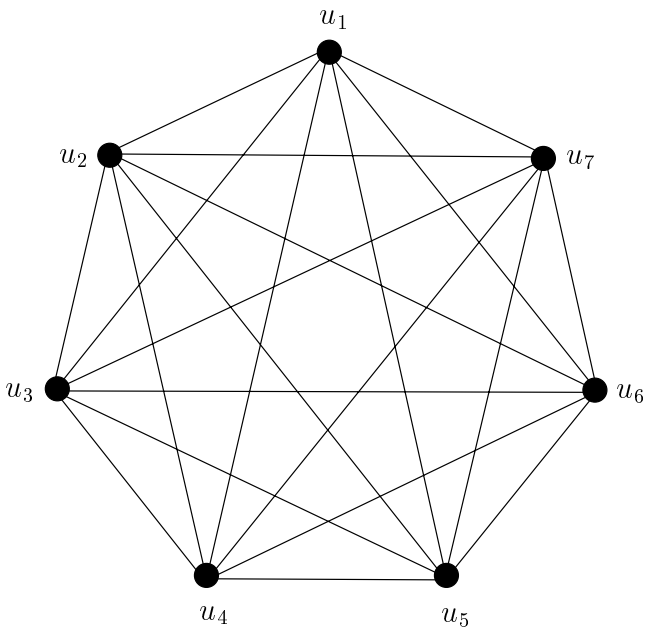


Figure 1.14: Complete graph K_7.

➤ **Note 1.21**
In this book, a graph is a finite graph with possibly some loops and multiple edges. In many places in the literature simple graphs are called graphs, and general graphs are called multigraphs *referring to their multiple edges.*

When drawing a graph, it is immaterial whether the lines are drawn straight or curved, long or short: the incidence between the edges and the vertices is of

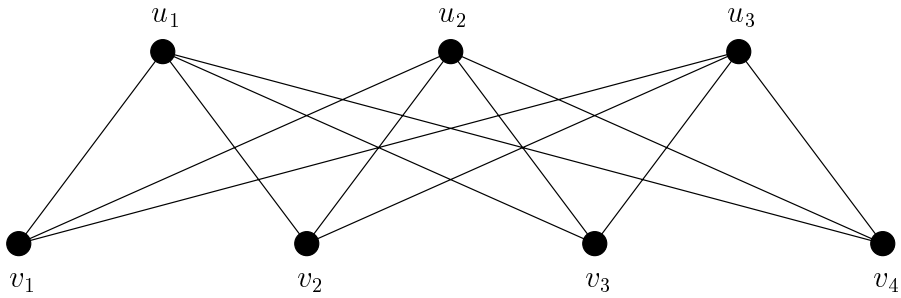


Figure 1.15: Complete bipartite graph $K_{3,4}$.

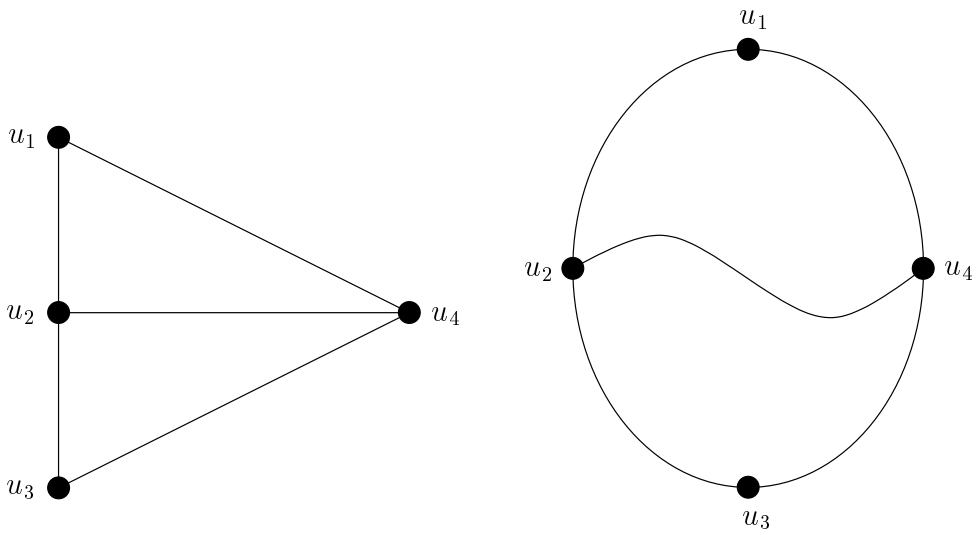


Figure 1.16: Same graph drawn in two ways.

great importance. For example, the two graphs drawn in Figure 1.16 are the same because incidence between vertices and edges is the same in both cases.

As we have already seen in the examples of diagrams of graphs, sometimes two edges intersect at a point that does not represent a vertex. Another example of this phenomenon occurs between edges e and f in Figure 1.17. Such edges should be thought of as being in different planes and thus having no common point. (Some authors break one of the two edges at such a crossing to emphasize this fact.)

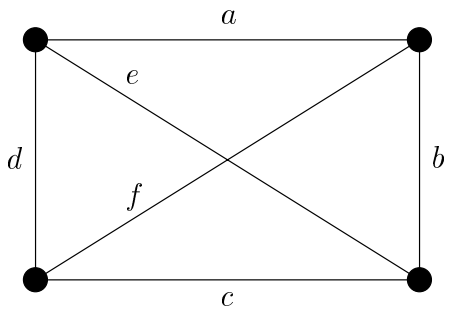


Figure 1.17: Edges e and f have no common point despite apparent intersection.

1.6 Degrees and Regular Graphs

Many properties of graphs can be stated in terms of numerical values that are attached to each vertex or to each edge of the graph by some logical rule. One of the most important of these attributes is the *degree* of a vertex that we now define.

Definition 1.22
Let $G = (V, E, \phi)$ be a graph and $u \in V$ a vertex of G. The degree *of u, denoted $d_G(u)$ or $d(u)$, is defined to be the number of edges having u as an endvertex. Note, the loops at u are counted twice. Formally,*

$$d(u) = |\{e \in E : u \in \phi(e), |\phi(e)| = 2\}| + 2|\{e \in E : u \in \phi(e), |\phi(e)| = 1\}|.$$

Informally speaking, $d(u)$ just tallies the number of "edge ends" connected to u. Thus it makes intuitive sense to count the loops twice since both loop's ends go into its endvertex. If G is a simple graph, then $d(u)$ is precisely the number of edges with u as an endvertex.

☞ **Example 1.23**
Consider the graph shown in Figure 1.18.

In this case we have

$$\begin{aligned} d(u_1) &= 3, \\ d(u_2) &= 2, \\ d(u_3) &= 4, \text{ and} \\ d(u_4) &= 3. \end{aligned}$$

We can now state our first theorem in graph theory. It relates degrees of vertices to the number of vertices and edges of a given graph.

Theorem 1.24 (Hand Shaking Theorem)
For a graph G we have

$$\sum_{u \in V(G)} d(u) = 2|E(G)|.$$

The name "Hand Shaking Theorem" comes from the following party analogy. Consider a collection of guests at a party. Suppose some guests shook hands with some other guests. If we asked everyone at the party how many guests they shook hands with and added those numbers all up, then this sum would be

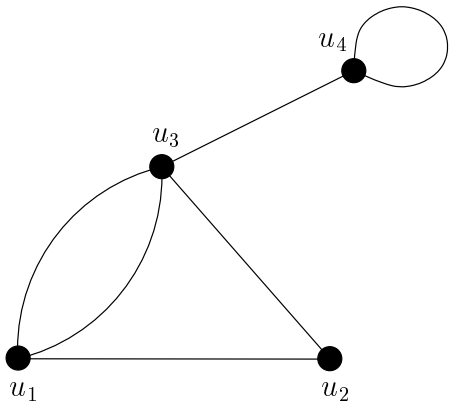


Figure 1.18: Various degrees.

equal to twice the number of total hand shakes. In graph theory terms, each vertex represents a guest and an edge between two guests represents a hand shake between them.

PROOF: (Theorem 1.24) In this proof we represent a loop in a graph by an edge $\{u, u\}$. Each edge $\{u, v\} \in E(G)$ contributes two to the sum on the lefthand side of the formula displayed in the theorem—one to $d(u)$ and one to $d(v)$. □

As a consequence of the Hand Shaking Theorem, we obtain the following corollary.

Corollary 1.25
In any graph the number of vertices of odd degree is always even.

PROOF: By the Hand Shaking Theorem, $\sum_{u \in V(G)} d(u)$ is always even. The number of odd $d(u)$-terms is therefore even. □

Graphs that have a uniform property often play an important role in applications. The following notion is one the most important such uniformity properties.

Definition 1.26
A graph G is called k-regular *if $d(u)$ equals k for all $u \in V(G)$. A graph G is* regular *if it is k-regular for some natural number k.*

Here are several examples of k-regular graphs.

☞ **Example 1.27**
We consider some typical regular graphs. Let $m, n \in \{1, 2, \ldots\}$ *unless otherwise noted.*

- ▷ *The null graph* N_n *is a 0-regular graph.*
- ▷ *For* n *a natural number greater than or equal to two, the cycle* C_n *is a 2-regular graph since* $d(u)$ *equals two for all* $u \in C_n(V)$.
- ▷ *The complete graph* K_n *is an* $(n-1)$*-regular graph.*
- ▷ *The complete* (m, n)*-bipartite graph* $K_{m,n}$ *is a regular graph if and only if* m *equals* n*, in which case* $K_{m,n}$ *is* n*-regular.*
- ▷ *The graph shown in Figure 1.9 is not regular since* $d(u_1) = 3 \neq 0 = d(u_6)$.

By the Hand Shaking Theorem, we obtain the following corollary about regular graphs.

Corollary 1.28
Every k*-regular graph on* n *vertices has* $kn/2$ *edges. In particular, the complete graph* K_n *has* $(n-1)n/2$ *edges.*

1.7 Subgraphs

When studying a specific graph, many times our attention is focused solely on a special part of the graph. Perhaps on a smaller graph lying inside the larger graph. This motivates the following definition of a *subgraph*.

Definition 1.29
For graphs $G' = (V', E', \phi')$ *and* $G = (V, E, \phi)$ *we say that* G' *is a* subgraph *of* G *if*

1. $V' \subseteq V$,
2. $E' \subseteq E$, *and*
3. $\phi'(e) = \phi(e)$ *for all* $e \in E$.

We denote the subgraph relation on graphs using the standard subset notation, namely $\subseteq$. If G' is a subgraph of G, we write $G' \subseteq G$.

In particular, for simple graphs $G' = (V', E')$ and $G = (V, E)$ we have that G' is a subgraph of G if $V' \subseteq V$ and $E' \subseteq E$.

Sometimes we need to talk about how a subgraph is derived from a group of vertices or edges. The following definition is useful for doing this.

Definition 1.30
Let G be a graph. Let $W \subseteq V(G)$ with $W \neq \emptyset$. The subgraph of G induced by *W, denoted $G[W]$, is the graph with vertex set W and edge set consisting of all the edges of G having both endvertices in W.*

Let $F \subseteq E(G)$ with $F \neq \emptyset$. The subgraph of G induced by *F, denoted $G[F]$, is the graph with edge set F whose vertex set consists of all the endvertices of edges of F.*

The following example illustrates this definition.

☞ **Example 1.31**
Consider the graph G depicted in Figure 1.19(a). In Figure 1.19(b) we present a subgraph G' of G. In Figure 1.19(c) we depict a subgraph induced by the vertex set $\{u_1, u_2, u_3\}$. That is, we show $G[\{u_1, u_2, u_3\}]$. In Figure 1.19(d) we show the subgraph induced by the edges $\{e_1, e_2, e_3, e_4\}$ together with their endvertices. That is, we show $G[\{e_1, e_2, e_3, e_4\}]$.

The next mathematical definition provides us with a mechanism for relating various graphs to one another.

Definition 1.32
Let S be a set. A relation $\preceq$ between the elements of S satisfying the following properties is called a partial order *on S:*

1. *It holds that $s \preceq s$ for all $s \in S$ (*reflexive *condition),*
2. *If $s \preceq t$ and $t \preceq u$, then $s \preceq u$ (*transitive *condition), and*
3. *If $s \preceq t$ and $t \preceq s$, then $s = t$ (*antisymmetric *condition).*

The set S together with a partial order $\preceq$ is called a partially ordered set *or* poset *for short.*

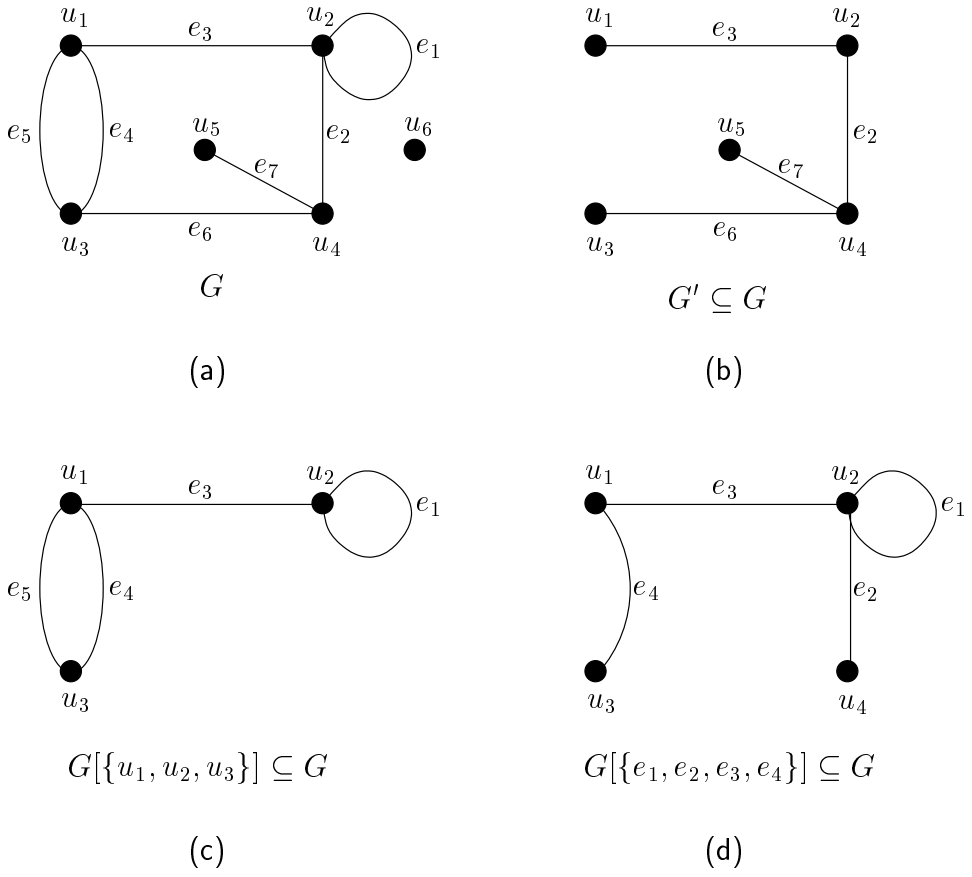


Figure 1.19: Various subgraphs.

Notice that the following properties hold for graphs and the subgraph relation:

1. For any graph G we have that $G \subseteq G$. Since each graph is a subgraph of itself, the subgraph relation satisfies the reflexive property.
2. If $G' \subseteq G''$ and $G'' \subseteq G$ then $G' \subseteq G$. So, the subgraph relation satifies the transitive condition.
3. If $G \subseteq G'$ and $G' \subseteq G$ then $G = G'$. So, the subgraph relation satisfies the antisymmetric condition.

This simply means that we have a partial order on any collection of subgraphs of a given graph. Some subgraphs are "larger" than others. Suppose G' and

G'' are both subgraphs of a graph G. It is possible that $G' \subseteq G''$ but not necessary. They may be incomparable using the subgraph relation. Hence, the word "partial" is used for this type of ordering.

Observation 1.33
Any collection of subgraphs of a given graph along with the $\subseteq$ relation forms a poset. Hence the relation "A is a subgraph of B" forms a poset.

1.8 Problem Solving Using Graph Theory

In what follows we illustrate how a particular problem can be solved using graphs. Similar techniques can be used to model and solve other problems. After some practice, it will become easier for you to model a wide range of problems with graphs.

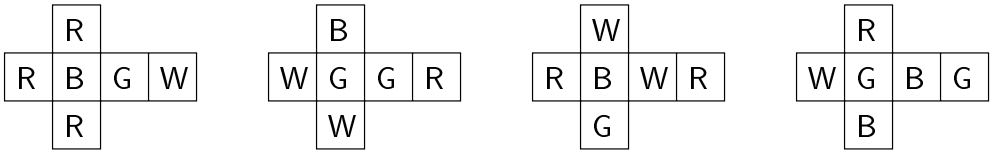
✍ **Problem 1.34**
A Puzzle with Multicolored Cubes

Two steps are involved in modeling the puzzle. First, the physical problem is converted into a problem of graph theory. Second, the graph-theory problem is solved. Let us consider the following well-known puzzle available in toy stores under the name of Instant Insanity.

We are given four cubes. The six faces of every cube are either colored blue, green, red, or white. Is it possible to stack the cubes one on top of the other to form a column such that no color appears twice on any of the four sides of this column? Clearly, a trial-and-error method is unsatisfactory because there are $41,472 = 3 \times 24 \times 24 \times 24$ possibilities.

Solution:

1. *Draw a graph with four vertices B, G, R, and W—one for each color. Pick a cube and call it cube 1; then represent its three pairs of opposite faces by three edges drawn between the vertices with appropriate colors. In other words, if a blue face in cube 1 has a white face opposite to it draw an edge between vertices B and W in the graph. Do the same for the remaining two pairs of faces in cube 1. Put label 1 on all three edges resulting from cube 1. A loop with the edge labeled 1 at vertex R, for instance, would result if cube 1 had a pair of opposite faces both colored red. Repeat the*



Cube 1 Cube 2 Cube 3 Cube 4

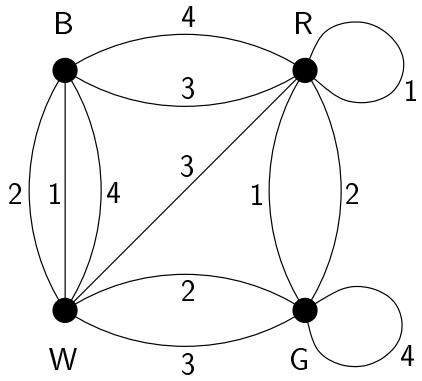


Figure 1.20: Four cubes unfolded and the corresponding color graph.

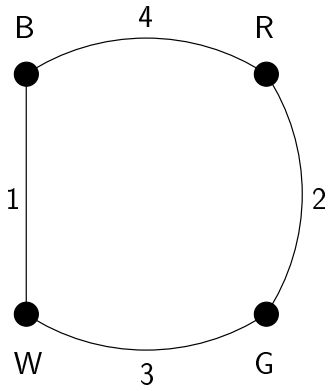
procedure for the other three cubes on these same vertices. The result is a graph with four vertices and 12 edges. A particular set of four colored cubes and its graph are shown in Figure 1.20.

2. *We analyze the graph resulting from this representation in what follows. The degree of each vertex is the total number of faces with the corresponding color. For the cubes of Figure 1.20, we have five blue faces, six green, seven red, and six white.*

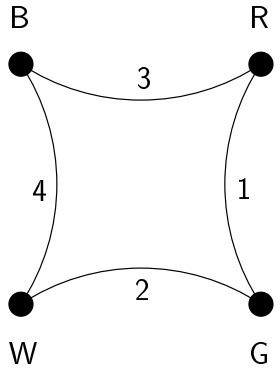
   Consider two opposite vertical sides of the desired column of four cubes, say facing north and south. A subgraph with four edges represents these eight faces—four facing north and four south. Each of the four edges in this subgraph has a different label—1, 2, 3, and 4. Moreover, no color occurs twice in either the north side or south side of the column if and only if every vertex in this subgraph is of degree two.

   Exactly the same argument applies to the east and west sides of the column.

   Thus the four cubes can be arranged to form a column such that no color



(a) North-South subgraph



(b) East-West subgraph

Figure 1.21: Two edge-disjoint subgraphs of the graph in Figure 1.20.

appears more than once on any side if and only if there exist two edge-disjoint subgraphs each having four uniquely labeled edges such that each vertex is of degree two. For the set of cubes shown in Figure 1.20 this condition is satisfied and the two subgraphs are shown in Figure 1.21.

Techniques similar to those used in Problem 1.34 can be applied to solve more complex problems as well. The key step is to determine how to obtain a visual representation of the problem using graphs. This is usually followed by an analysis phase.

1.9 A Brief History of Graph Theory

This section is devoted to a brief glimpse of graph theory's history. Additional information can be obtained by pursing the references.

As mentioned before, graph theory was born in 1736 with Leonhard Euler's paper in which he solved the Königsberg bridge problem. For the next 100 years nothing more was done in the field.

In 1847, Gustav Robert Kirchhoff (1824-1887) developed the theory of trees for their applications in electrical networks. Ten years later, Arthur Cayley (1821-1895) discovered trees while he was trying to enumerate the isomers of saturated hydrocarbons C_nH_{2n+2}. About the time of Kirchhoff and Cayley, two other

important problems in graph theory were formulated. One was the *four-color problem*. It asks whether four colors are sufficient for coloring any atlas (a map drawn in the plane) so that countries with common boundaries have different colors.

It is believed that August Ferdinand Möbius (1790-1868) first presented the four-color problem in one of his lectures in 1840. About ten years later, Augustus De Morgan (1806-1871) discussed this problem with his fellow mathematicians in London. De Morgan's letter is the first authenticated reference to the four-color problem. The problem became well-known after Cayley published it in 1879 in the first volume of the *Proceedings of the Royal Geographic Society*. The four-color problem was finally solved by Kenneth Appel and Wolfgang Haken in 1976; and hence, it is now justifiably called the *Four Color Theorem*. Appel and Haken's proof depends on extensive computer calculation to check 1476 nontrivial cases. These are the so-called "unavoidable configurations."

A new approach was given to the proof of the Four Color Theorem in the 1990's by Neil Robertson, Daniel Sanders, Paul Seymour, and Robin Thomas where the number of cases of unavoidable configurations was reduced to 633 (see [12]). Interestingly, they found the computer program worked faster when dealing with edge colorings instead of vertex colorings. This was good since a famous equivalence relation of Peter Guthrie Tait (1831–1901) from 1878 states that every planar graph satisfying certain conditions is edge 3-colorable if and only if it is vertex 4-colorable. The number of cases one needs to check is still too huge for one person to verify. Hence, the proof still depends extensively on computer computations. Therefore, many mathematicians and computer scientists remain waiting and hoping for a self-contained and explanatory proof of the Four Color Theorem.

The other important problem formulated during the time of Kirchhoff and Cayley was due to Sir William Rowan Hamilton (1805-1865). In the year 1859, he invented a puzzle and sold it for 25 guineas to a game manufacturer in Dublin. The puzzle consisted of a wooden, regular dodecahedron (a polyhedron with 12 faces and 20 corners, each face being a regular pentagon and three edges meeting at each corner; see Figure 1.22). The corners were marked with the names of 20 important cities: Delhi, London, New York, Paris, and so on. The object in the puzzle was to find a route along the edges of the dodecahedron passing through each of the 20 cities exactly once. In Chapter 3 we will see that the solution of this specific problem is easy to obtain. However, no one has found a necessary

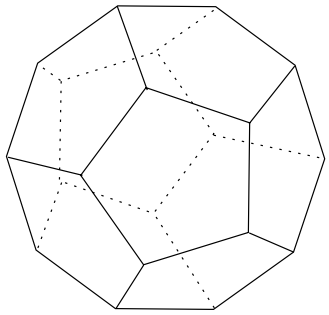


Figure 1.22: Dodecahedron.

and sufficient condition for the existence of such a route, called a *Hamiltonian cycle*, in an arbitrary graph.

This fertile period was followed by half a century of relative inactivity. Then a resurgence of interest in graphs started during the 1920s. One of the pioneers in this period was Denes König (1884–1944). He organized the work of other mathematicians with his own research, and wrote the first book on the subject. It was published in 1936.

The past 30 years has been a period of intense activity in graph theory—both applied and pure. A great deal of research continues to be done. Many papers have been published and numerous books on graph theory have been written during the past decades. Among the most interesting characters who have worked in the field, the wandering mathematician Paul Erdös (1913–1996) with his many contributions to graph theory ranks near the top.

1.10 Remainder of This Book

We highlight the contents of each of the remaining chapters below.

In Chapter 2 we look at paths and cycles, and examine the concepts of connectedness, homomorphisms, and isomorphisms.

Chapter 3 contains Eulerian graphs, several important operations on graphs, bipartite graphs, Hamiltonian paths and cycles, and a look at the well-known traveling salesman problem.

In Chapter 4 we present the concepts of trees and forests, demonstrate why

trees are one of the most important types of graphs, discuss the fundamental properties of trees and forests, present the concept of rooted trees, define binary and regular binary trees, present inductive proofs on trees, examine some important applications of trees, prove the Erdös-Szekeres Theorem, define distance and centers for graphs in general and specifically to trees, describe some characterizations of trees, and study path lengths of trees with respect to algorithmic efficiency.

Chapter 5 includes a classic application of trees to a problem in chemistry, the concepts of spanning tree and spanning forest, Cayley's Theorem for counting labeled trees, Prüfer codes and their uses, how to find all of the spanning trees of a graph, the minimum cost spanning tree, KRUSKAL'S and PRIM'S algorithms, several applications of spanning trees, and degree-constrained spanning trees.

In Chapter 6 we define edge cuts and discuss their basic properties, examine edge and vertex connectivity issues, define blocks and separability of a graph, discuss the structure of 1- and 2-connected graphs, and explore the notions of 1- and 2-isomorphisms for graphs.

Chapter 7 defines planar graphs, discusses graphs embeddable in various surfaces, presents Euler's Formula for planar graphs and various consequences of it, presents a characterization of planar graphs in the Kuratowski-Wagner Theorem, and defines homeomorphic graphs, contraction, and minors of graphs.

In Chapter 8 we examine geometric duals of planar graphs, define combinatorial duals of graphs, present characterizations of planar graphs in terms of duality and basic cycles, explore thickness and crossings as measures of the degree of planarity, define the genus of a general graph, and give a generalization of Euler's Formula.

In Chapter 9 we define semigroups, monoids, and groups, discuss rings and fields, present modular arithmetic and Galois fields, define a vector space over a field, define a vector space associated with a graph, study basis vectors of a graph, look at cycle and edge cut subspaces, and prove some graph-theoretic results using vector spaces.

In Chapter 10 we define the adjacency matrix for a graph, define the (reduced) incidence matrix, present the Matrix-Tree Theorem, define the (fundamental) circuit matrix, define the (fundamental) edge cut matrix, derive relationships among these matrices, present an application in a switching network, and present an application in electrical network analysis.

1.11 Exercises

1. Draw all simple graphs having exactly one, two, three, or four vertices.

2. Draw a portion of your academic department's Web page as a graph by modeling Web pages as vertices and hyperlinks as edges. Include at least ten different Web pages and draw edges for all of the existing hyperlinks between the Web pages you select. Label the vertices with their uniform resource locators.

3. Draw graphs representing problems of (a) two houses and the three utilities water, gas, and electricity (b) four houses and the four utilities water, gas, electricity, and telephone.

4. Model the seating assignments problem for seven club members. List a set of sequences that illustrates the maximum number of possible different seating arrangements satisfying the conditions of Problem 1.6.

5. Refer to the job assignments problem shown in Figure 1.6. Is the company able to meet its hiring needs? If so, provide a possible set of hires that meet their needs?

6. Name six situations (games, activities, real-life problems, and so on) that can be represented by a graph. Explain what the vertices and edges represent.

7. Draw a graph with 64 vertices representing the squares of a chess board. Join these vertices appropriately by edges, each representing a move of the knight. You will see that every vertex in this graph has degree two, three, four, six, or eight. How many vertices are there of each type?

8. Write out complete 3-tuples that describe the graphs shown in Figures 1.6 and 1.7.

9. Show that an infinite graph with a finite number of edges (that is, a graph with a finite number of edges and an infinite number of vertices) must have an infinite number of isolated vertices.

10. Show that an infinite graph with a finite number of vertices (that is, a graph with a finite number of vertices and an infinite number of edges) will have at least one pair of vertices or one vertex that is joined by an infinite number of edges.

11. Show that the maximum degree of any vertex in a simple graph with n vertices is $n-1$.

12. Show that the maximum number of edges in a simple graph with n vertices is $(n-1)n/2$.

13. For n equals one, two, and five draw the following graphs: N_n, P_n, C_n, K_n, and $K_{n,n}$. Which of the graphs are the same?

14. Construct a table whose entries are formulas for the number of vertices and edges for the graphs N_n, P_n, C_n, K_n, and $K_{n,n}$ for an arbitrary value of n.

15. For what values of n less than 20 can you draw a 3-regular graph having n vertices?

16. Express both the maximum and minimum degree of vertices in the graphs N_n, P_n, C_n, K_n, and $K_{n,n}$ in terms of n.

17. How are N_n, P_n, and C_n related via the subgraph relationship?

18. For what values of n is K_n a subgraph of $K_{l,m}$ for given values of l and m?

19. Specify a minimum value of n in terms of l and m such that $K_{l,m}$ is a subgraph of K_n.

20. For every k-regular graph is there a $(k+1)$-regular graph that contains the k-regular graph as a subgraph?

21. Prove that the usual less than relation $(<)$ is a partial order on the set of *integers* $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$.

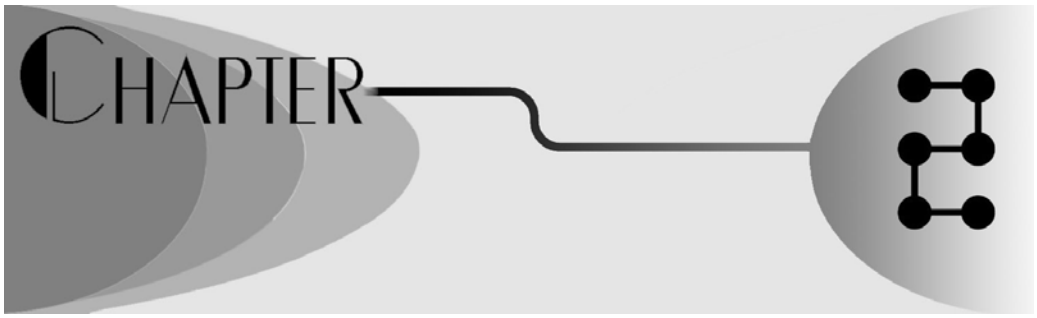
22. Define a relation on graphs that is a partial order and include a proof that your relation is indeed a partial order.

23. *Decanting problem.* You are given three vessels A, B, and C of capacities 8, 5, and 3 gallons respectively. A is initially filled while B and C are empty. Divide the liquid in A into two equal quantities. This is the classical decanting problem. [*Hint:* Let a, b, and c be the amounts of liquid in A, B, and C respectively. We have $a + b + c = 8$ at all times. Since at least one of the vessels is always empty or full at least one of the following equations must always be satisfied: $a = 0$, $a = 8$, $b = 0$, $b = 5$, $c = 0$, or $c = 3$. You will find that with these constraints there are 16 possible states (situations) in this process. Represent this problem by means of a 16-vertex graph. Each vertex stands for a state and each edge for a permissible change of states between its two endvertices. Now when you look at this graph, it will be clear to you how to go from state $(8, 0, 0)$ to state $(4, 4, 0)$.]

24. Refer to the set of cubes represented by the graph shown in Figure 1.20. Is it possible to stack all four cubes into a column such that each side shows only one color? Explain.

25. Write a one page biography of Paul Erdös. Be sure to include several of his most popular sayings and a description of the Erdös number. [*Hint:* Use the World Wide Web to conduct your research.]

26. Construct a timeline that includes twenty-five of the most important events in graph theory. [*Hint:* Use the World Wide Web to conduct your research.]



Basic Concepts in Graph Theory

Chapter Goals

Define and study paths and cycles.	Present homomorphisms.
Examine connectivity.	Present isomorphisms.

2.1 Introduction

In this chapter we continue to develop some basic tools to handle problems in graph theory. These building blocks are essential in order to discuss more complex parts of this subject. In §2.2 we present paths and cycles. These ideas lead us to the notion of connectivity, the topic of §2.3. In §2.4 we present two interesting facts about graph components. We then examine the structures of graphs. The concepts of homomorphisms and isomorphisms introduced in §2.5 and §2.6 allow us to compare graphs and determine their similarities.

2.2 Paths and Cycles

Although two vertices in a given graph may not be adjacent, they might be adjacent to a common neighbor, or more generally, connected by a sequence of

edges. This type of connectivity turns out to be an important concept in graph theory. The following definition captures this idea.

Definition 2.1

▷ *A* walk *in a graph G is an alternating sequence*

$$(u_0, e_1, u_1, e_2, \ldots, e_k, u_k)$$

of vertices and edges that begins and ends with a vertex. For each $i \in \{1, 2, \ldots, k\}$ the endvertices of e_i are u_{i-1} and u_i. That is,

$$\phi(e_i) = \{u_{i-1}, u_i\}.$$

▷ *The vertex u_0 is the* initial vertex *of the walk. The vertex u_k is the* final vertex *of the walk. The initial or final vertex of a walk is also called an* endvertex *of the walk.*

▷ *The natural number k is the* length *of the walk.*

▷ *A* trail *in G is a walk with all of its edges $e_1, e_2, \ldots, e_k$ distinct.*

▷ *A* path *in G is a walk with all of its vertices $u_0, u_1, \ldots, u_k$ distinct.*

▷ *For vertices u and v in G, a u, v-*walk *(*u, v-trail *or* u, v-path*) is a walk (respectively, trail or path) with initial vertex u and final vertex v.*

▷ *A walk or trail of length at least one is* closed *if the initial vertex and the final vertex are the same. A closed trail is also called a* circuit.

▷ *A* cycle *is a closed trail with distinct vertices except the initial and final vertices are the same.*

Note that one can view a single vertex as a walk, trail, or path of length 0. We extend Definition 2.1 to include these cases. Cycles always have positive length. We never view a single vertex as a trivial cycle. The only cycles of length one are loops. Also, the set of vertices and edges constituting a given walk, trail, path, or cycle in a graph G clearly forms a subgraph of G.

Observe that in a simple graph a walk, trail, path, or cycle can be specified by a sequence of vertices $(u_0, u_1, \ldots, u_k)$ instead of $(u_0, e_1, u_1, e_2, \ldots, e_k, u_k)$. This is because a pair of adjacent vertices u_{i-1} and u_i completely determine the only edge $e_i = \{u_{i-1}, u_i\}$ between them.

The next example illustrates Definition 2.1.

☞ **Example 2.2**
Consider the graph shown in Figure 2.1.

- ▷ $w = (u_1, a, u_2, b, u_3, c, u_3, g, u_1, a, u_2, e, u_4, h, u_5)$

  In the figure we see w is a walk of length seven. Note that w is not a trail since the edge a appears twice. A walk that is not a trail is clearly not a path. Hence, w is not a path.

- ▷ $t = (u_1, a, u_2, b, u_3, c, u_3, d, u_4, e, u_2, f, u_5)$

  The walk t is a trail of length six. Note that t is not a path since the vertices u_2 and u_3 appear twice.

- ▷ $p = (u_1, a, u_2, b, u_3, d, u_4)$

  The walk p is a path of length three.

- ▷ $c_1 = (u_1, a, u_2, b, u_3, c, u_3, d, u_4, e, u_2, a, u_1)$

  The walk c_1 is a closed walk of length six.

- ▷ $c_2 = (u_1, g, u_3, c, u_3, b, u_2, e, u_4, h, u_5, f, u_2, a, u_1)$

  The walk c_2 is a closed trail of length seven.

- ▷ $c_3 = (u_1, g, u_3, d, u_4, e, u_2, a, u_1)$

  The walk c_3 is a cycle of length four.

Sometimes we need to combine walks, trails, or paths. Clearly, if we know how to go from A to B and B to C, then we can get from A to C. This observation inspires the following definition.

Definition 2.3
Let p and q be two walks, two trails, or two paths with

$$p = (u_0, e_1, u_1, \ldots, e_k, u_k) \text{ and}$$
$$q = (v_0, f_1, v_1, \ldots, f_l, v_l).$$

If $u_k = v_0$ then we can form the composite *walk, trail, or path*

$$pq = (u_0, e_1, u_1, \ldots, e_k, u_k, f_1, v_1, \ldots, f_l, v_l)$$

by first traversing along p and then along q.

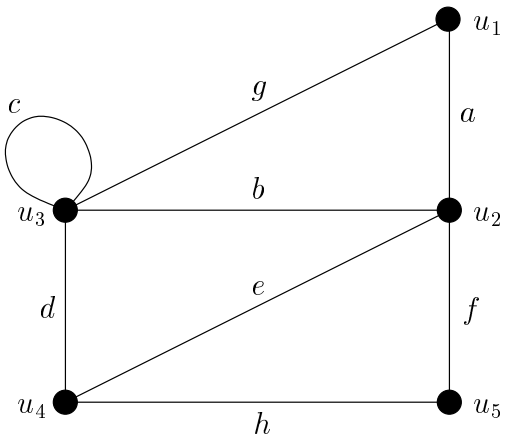


Figure 2.1: Graph used to illustrate walks, trails, paths, and cycles.

We can also form the inverse *walk, trail, or path of* p

$$p^{-1} = (u_k, e_k, \ldots, u_1, e_1, u_0).$$

We observe that although the concept of path and cycle are very simple, their formal definitions become quite involved. The complications arise in specifying precisely which vertices can and can not be duplicated.

Note the "left-right" difference between compositions of walks, trails, or paths and the composition of two maps g and f. The composition of maps $(g \circ f)(x)$ means that we first *apply* f to x on its right, and then g to $f(x)$ on its right. Walks, trails, or paths are not applied to anything on their right. Instead they are simply concatenated.

The diagram in Figure 2.2 explains the relationship of the items from Definition 2.1 in any given graph G. In the figure an arrow from one oval to another means that items in the first oval are more general than items in the second. So, for example, walks are a subset of trails.

The diagram in Figure 2.2 is actually an example of a *directed graph*, where each edge has a direction. We will discuss directed graphs in more detail in Chapter 13.

The next result shows that if there is a walk between two vertices in a graph, then there must also be a path between them. Intuitively, the idea is simply to go directly between the two vertices without taking any "detours."

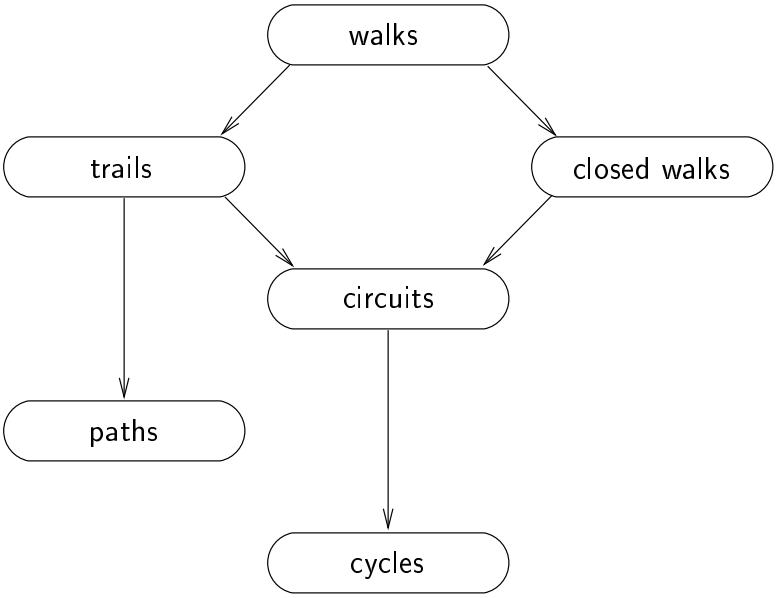


Figure 2.2: Relationship between various subgraphs and walks.

Theorem 2.4
Let G be a graph having distinct vertices u and v. If G contains a u, v-walk, then G contains a u, v-path.

PROOF: If there is a u, v-walk in G, then there is one of shortest length.

Let p be a u, v-walk of shortest length k:

$$p = (u_0, e_1, u_1, e_2, \ldots, e_k, u_k),$$

where $u = u_0$ and $v = u_k$.

We want to show that p is actually a path. Assume the contrary, that p is not a path. Then there must be a repeated vertex in p. That is, there must be a pair of indices $i, j \in \{0, 1, 2, \ldots, k\}$ with $i < j$ and $u_i = u_j$. In this case we can form a new u, v-walk

$$p' = (u_0, e_1, u_1, \ldots, e_i, u_i, e_{j+1}, u_{j+1}, \ldots, e_k, u_k)$$

of length $k - (j - i) < k$. Figure 2.3 depicts the situation. This means that p is not the shortest u, v-walk. Thus we have a contradiction. Our assumption that p is not a path is therefore wrong. This proves the theorem. □

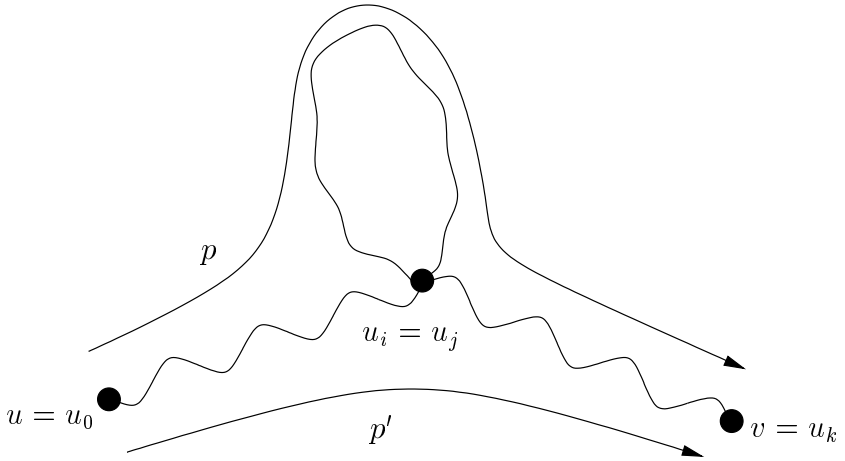


Figure 2.3: Walk p' is *shorter* than walk p.

It should be clear from context when we talk about walks, trails, paths, or cycles whether we mean the actual sequence of vertices and edges as defined, or simply the underlying subgraph it produces. The phrase "G contains a trail of length five" means that the graph G contains a subgraph with five edges and their endvertices such that this very subgraph makes up a trail of length five in an unambiguous and clear fashion.

2.3 Connectedness

Intuitively, the concept of *connectedness* is obvious. A graph is connected if we can reach any vertex from any other vertex by traveling along the edges. This leads to the following definition.

Definition 2.5
A graph G is said to be connected *if for every pair of vertices, u and v in G, the graph has a u, v-path. Otherwise, we say the graph is* disconnected.

The next example illustrates this definition.

☞ **Example 2.6**
Consider the graphs G and G' depicted in Figure 2.4. G is a connected graph but G' is not since G' has no u, v-path.

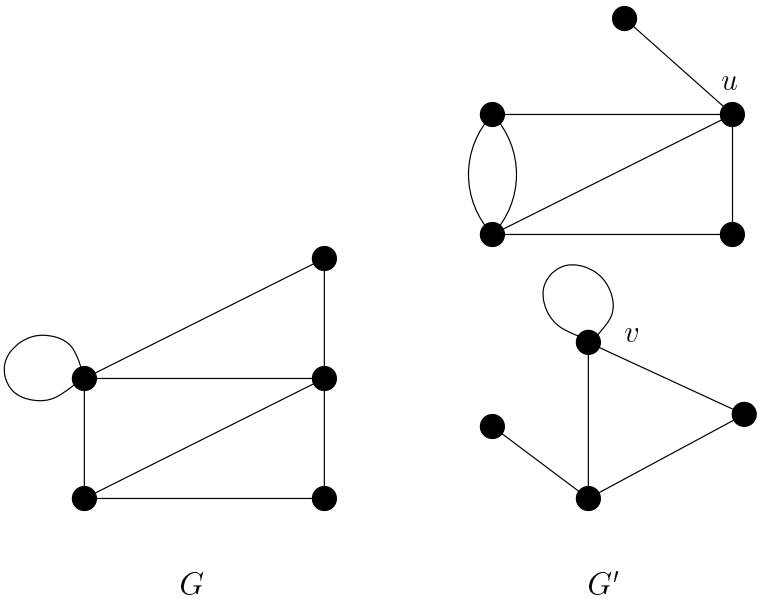


Figure 2.4: Graph G is connected but G' is not.

It is not hard to see that a disconnected graph consists of two or more connected subgraphs. Each of these connected subgraphs is called a *component* or *connected component* of the graph. More formally, we have the following definition.

Definition 2.7
Let G be a graph. Let $C_1, \ldots, C_k$ be subgraphs of G whose vertex sets and edge sets are pairwise disjoint and such that they cover all the vertices and edges of G. That is,

$$V(G) = V(C_1) \cup \cdots \cup V(C_k) \text{ and}$$
$$E(G) = E(C_1) \cup \cdots \cup E(C_k),$$

where $V(C_i) \cap V(C_j) = E(C_i) \cap E(C_j) = \emptyset$ for each distinct i and j.

Each of the subgraphs C_i is called a component *or* connected component *of G. The* order *of a component C_i is simply the number of vertices in C_i.*

Note Definitions 1.9 on page 13 and 1.29 on page 22 imply that $V(C_i) \neq \emptyset$ for each i since each C_i is a subgraph. The next result states the intuitive idea that every graph is made up of its components and its components alone.

Theorem 2.8
Every graph G has a finite number of connected components $C_1, \ldots, C_k$.

It is evident that a graph G is connected if and only if k is equal to one.

Before proving Theorem 2.8, we define a very convenient mathematical tool. This notion has far reaching applications.

Definition 2.9
Let S be a set. A relation $\sim$ between the elements of S that satisfies the following properties is called an equivalence relation *of S:*

1. *For all $s \in S$, $s \sim s$ (*reflexive *condition),*
2. *If $s \sim t$ then $t \sim s$ (*symmetric *condition), and*
3. *If $s \sim t$ and $t \sim u$, then $s \sim u$ (*transitive *condition).*

Shortly, we will see an example of a graph-theoretic equivalence relation.

The following result shows how an equivalence relation can be used to split its underlying set into pieces. This type of division is called a *partition*.

Theorem 2.10
Let X be a finite set and $\sim$ an equivalence relation on X. Then we have a partition

$$X = X_1 \cup \cdots \cup X_k$$

for some natural number k, where

1. *X_i is nonempty for $1 \leq i \leq k$,*
2. *$X_i \cap X_j = \emptyset$ for each distinct i and j between 1 and k, and*
3. *$X_i = \{y \in X : y \sim x_i\}$ for every $x_i \in X_i$.*

Each of the subsets X_i is called an equivalence class *corresponding to $\sim$.*

PROOF: (Sketch) Choose $x_1 \in X$ and let $X_1 = \{y \in X : y \sim x_1\}$. If X_1 equals X then we stop. Otherwise, we pick $x_2 \in X \setminus X_1$ and let $X_2 = \{y \in X : y \sim x_2\}$. Note that because $\sim$ is an equivalence relation, we see that

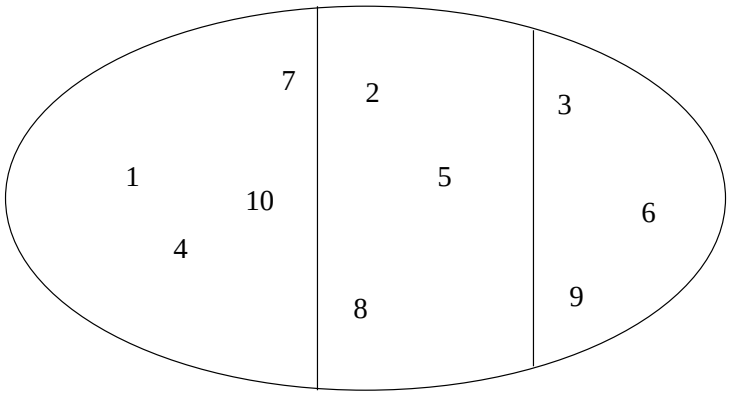


Figure 2.5: Partition of a set $S = \{1, 2, \ldots, 10\}$ into three equivalence classes. (The area of each region is not significant.)

$X_1 \cap X_2 = \emptyset$. Since otherwise, if $y \in X_1 \cap X_2$, then we have $x_1 \sim y$ and $x_2 \sim y$. Hence, by the symmetric and transitive conditions, we get $x_1 \sim x_2$. But then $x_2 \in X_1$, which is not the case.

Again, if $X = X_1 \cup X_2$, then we stop. Otherwise, we continue in this manner and pick $x_3 \in X \setminus (X_1 \cup X_2)$ and let $X_3 = \{y \in X : y \sim x_3\}$. As before, we have $X_3 \cap X_i = \emptyset$ for i equals 1 or 2.

The procedure outlined here must stop because X is a finite set. At the stopping point, we have that $X = X_1 \cup \cdots \cup X_k$, where $x_1, \ldots, x_k \in X$ and $X_i \cap X_j = \emptyset$ for all distinct i and j between 1 and k. Since $x_i \in X_i$ for $1 \leq i \leq k$, we see that X_i is nonempty.

Finally, let $i \in \{1, \ldots, k\}$ and let $x_i' \in X_i$. Since $x_i' \sim x_i$, we have that if $y \sim x_i$ then $y \sim x_i'$ and the other way, that if $y \sim x_i'$ then $y \sim x_i$. Therefore,

$$X_i = \{y \in X : y \sim x_i\} = \{y \in X : y \sim x_i'\}.$$

This holds for each $x_i' \in X_i$ and we have the theorem. □

Figure 2.5 depicts the partition of the set $S = \{1, 2, \ldots, 10\}$ corresponding to the relation *congruent to the same value mod three*. Notice S is partitioned into three equivalence classes. For example, $3 \bmod 3 = 0$, $6 \bmod 3 = 0$, and $9 \bmod 3 = 0$, so these three values all fall into the same part (equivalence class) of the partition.

We need one more result before tackling the proof of Theorem 2.8. Let G be a graph. We define the *path relation* between the vertices of G as follows:

$$u \sim v \text{ if and only if there is a } u,v\text{-path in } G. \tag{2.1}$$

The next theorem shows that the path relation is an equivalence relation.

Theorem 2.11
Let G be a graph. The path relation is an equivalence relation between the vertices of G. That is,

1. *For all $u \in V(G)$, $u \sim u$,*
2. *If $u \sim v$ then $v \sim u$, and*
3. *If $u \sim v$ and $v \sim w$, then $v \sim w$.*

PROOF: For any vertex u the path of length zero is in G and hence $u \sim u$.

Suppose $u \sim v$ and that p is a u,v-path in G. The inverse p^{-1} is also a v,u-path in G and hence $v \sim u$.

Finally, if p is a u,v-path in G and q is a v,w-path in G, then pq is a u,w-walk in G. Hence, by Theorem 2.4 there is a u,w-path in G. Thus we have the theorem. □

We are now ready to present the proof of Theorem 2.8.

PROOF: (Theorem 2.8, page 41) By Theorem 2.11 using the path relation, we get a partition of the vertices of G

$$V(G) = V_1 \cup \cdots \cup V_k,$$

where $V_i \cap V_j = \emptyset$ for all distinct i and j between 1 and k. Note that every pair of vertices u and v in each V_i is connected by a path in G.

Let $C_i = G[V_i]$ be the subgraph of G induced by V_i. Let E_i be the edge set of C_i for i between 1 and k. Since no two of the C_i's have a vertex in common, they can have no edge in common either. Hence, $E_i \cap E_j = \emptyset$ for all distinct i and j between 1 and k.

Finally, if $e \in E$ is an edge of G, then the endvertices of e say u and v are connected by the path (u,e,v). So, $u,v \in V_i$ for some i. Therefore, $e \in E_i$. Thus we have the covering

$$E(G) = E_1 \cup \cdots \cup E_k.$$

This proves the theorem. □

2.4 Two Facts about Graph Components

In this section we present two interesting facts about graphs and their components.

Theorem 2.12
If a graph G has exactly two vertices u and v of odd degree, then G has a u, v-path.

PROOF: Assume that all of the vertices of G except u and v are of even degree. If u and v are both in the same component of G, then that component of G and hence G itself has a u, v-path. Otherwise, u and v are in different components. Let C_u and C_v be the respective components. The only odd degree vertex of C_u (C_v) is u (respectively, v). This contradicts Corollary 1.25 on page 21. □

Before proving our next theorem, we need an algebraic inequality.

Observation 2.13
If $x_1, x_2, \ldots, x_k \geq 1$ are real numbers and $x = x_1 + \cdots + x_k$, then

$$\sum_{i=1}^{k} x_i^2 \leq x^2 - (k-1)(2x-k).$$

PROOF: For each i let $x_i = y_i + 1$, where $y_i \geq 0$. Let $y = y_1 + \cdots + y_k$. By expressing the right hand side of the above inequality in terms of y, we have

$$\begin{aligned} x^2-(k-1)(2x-k) &= (y+k)^2-(k-1)(2y+k) \\ &= y^2+2y+k \\ &= \left(\sum_{i=1}^{k} y_i\right)^2 + 2\left(\sum_{i=1}^{k} y_i\right) + k \\ &\geq \sum_{i=1}^{k} y_i^2 + 2\left(\sum_{i=1}^{k} y_i\right) + k \\ &= \sum_{i=1}^{k} (y_i^2+2y_i+1) \\ &= \sum_{i=1}^{k} x_i^2. \end{aligned}$$

This proves the observation. □

We now have the necessary machinery to prove the following theorem.

Theorem 2.14
A simple graph G with n vertices and k components can have at most

$$(n-k)(n-k+1)/2$$

edges.

PROOF: Let $C_1, \ldots, C_k$ be the components of G. Denote the order of C_i by n_i for $1 \leq i \leq k$. Note that $n = n_1 + \cdots + n_k$. Observe that each C_i is a simple graph. From Corollary 1.28 on page 22, it follows that C_i can have at most $(n_i-1)n_i/2$ edges. Hence, using Observation 2.13, the maximum number of edges of G can be estimated as follows:

$$\begin{aligned}\sum_{i=1}^{k}\frac{(n_i-1)n_i}{2} &= \frac{1}{2}\left(\sum_{i=1}^{k}n_i^2\right)-\frac{n}{2}\\ &\leq \frac{1}{2}\left[n^2-(k-1)(2n-k)\right]-\frac{n}{2}\\ &= \frac{(n-k)(n-k+1)}{2}.\end{aligned}$$

This proves the theorem. □

There are many other interesting facts about graph components. The two results presented in this section give a flavor of how such results are proven.

2.5 Homomorphisms and Isomorphisms of Graphs

Many mathematical objects can be studied and described by certain special maps between them, and the same holds true for graphs. These maps are called *homomorphisms*. A homomorphism from a graph G to another graph G' is a function that maps the vertices of G to the vertices of G', and the edges of G to the edges of G' in the following way. If an edge e in G has endvertices u and v, then the corresponding image edge e' of G' has endvertices u' and v'; these are the respective images of u and v under the mapping.

To state this precisely and to be in coherence with Definition 1.9 on page 13, we make the following definition.

Definition 2.15
Let $G=(V,E,\phi)$ and $G'=(V',E',\phi')$ be two graphs. A homomorphism

$$f : G \mapsto G'$$

from G to G' is an ordered pair $f=(f_1,f_2)$ of maps $f_1 : V \mapsto V'$ and $f_2 : E \mapsto E'$ satisfying the following condition:

$$\text{if } \phi(e)=\{u,v\} \text{ then } \phi'(f_2(e))=\{f_1(u),f_1(v)\}.$$

That is, if u and v are the endvertices of e in G, then $f_1(u)$ and $f_1(v)$ are the endvertices of $f_2(e)$ in G'.

If both maps f_1 and f_2 are bijective, then f is called an isomorphism*. In this case we say that the graph G is* isomorphic *to the graph G', or that G and G' are* isomorphic*. We denote graph isomorphism by*

$$G \cong G'.$$

An isomorphism from G to itself is called an automorphism *of G.*

Before discussing the implication of this definition further, we consider two examples. The first example illustrates the concept of homomorphism.

☞ **Example 2.16**
Consider the two graphs G and G' shown in Figure 2.6.

Define the maps f_1 and f_2 as follows:

$$\begin{aligned} &f_1(u_1) = f_1(u_4) = f_1(u_5) = f_1(u_6) = u'_4, \\ &f_1(u_2) = u'_2, \text{ and} \\ &f_1(u_3) = u'_3, \end{aligned}$$

and

$$\begin{aligned} &f_2(e_1) = f_2(e_7) = f_2(e_8) = e'_1, \\ &f_2(e_2) = e'_4, \\ &f_2(e_3) = e'_5, \\ &f_2(e_4) = f_2(e_5) = e'_6, \text{ and} \\ &f_2(e_6) = e'_7. \end{aligned}$$

Let $f : G \mapsto G'$ be defined by $f = (f_1, f_2)$. Having defined the maps f_1 and f_2, one can consider each edge in G and check whether its endvertices are mapped to the corresponding endvertices of the mapped edge in G'. So, we consider the list of edges $e_1, e_2, \ldots, e_8$:

▷ *We have $\phi(e_1) = \{u_1\}$ and $\phi'(f_2(e_1)) = \phi'(e'_1) = \{u'_4\} = \{f_1(u_1)\}$. So, this checks out okay.*

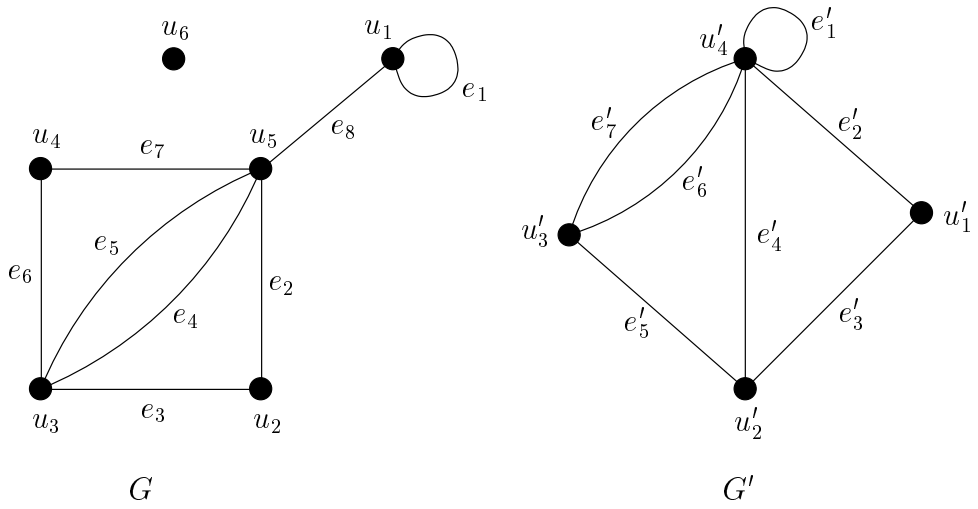


Figure 2.6: G and G' used in Example 2.16 to illustrate a homomorphism.

▷ *We have* $\phi(e_2) = \{u_2, u_5\}$ *and* $\phi'(f_2(e_2)) = \phi'(e'_4) = \{u'_2, u'_4\} = \{f_1(u_2), f_1(u_5)\}$. *So, this checks out okay.*

...

▷ *We have* $\phi(e_8) = \{u_1, u_5\}$ *and* $\phi'(f_2(e_8)) = \phi'(e'_1) = \{u'_4\} = \{u'_4, u'_4\} = \{f_1(u_1), f_1(u_5)\}$. *So, this checks out okay.*

Hence, $f : G \mapsto G'$ *is indeed a homomorphism. Note that* f *is not an isomorphism.*

The following remark helps clarify how one graph can be mapped to another by a homomorphism.

Remark 2.17
By the definition of homomorphism, any pair of adjacent vertices are mapped to adjacent vertices or to the same vertex. Moreover, any pair of edges incident with each other are mapped to edges that are incident with each other or to the same edge. Also a non-loop edge can be mapped to a non-loop edge or to a loop. However, a loop can only be mapped to a loop.

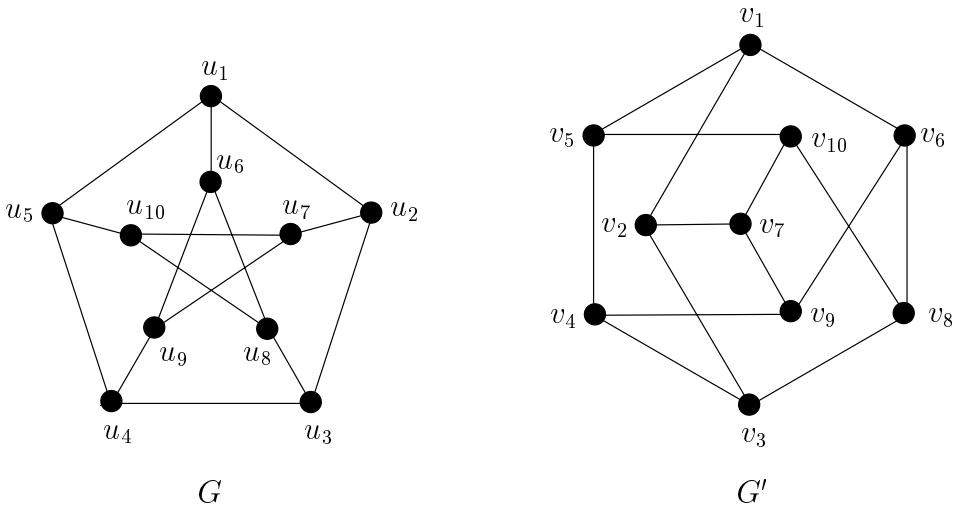


Figure 2.7: Are graphs G and G' isomorphic?

The following example illustrates the notion of isomorphism.

☞ **Example 2.18**
Consider the two graphs G and G' depicted in Figure 2.7.

Here both G and G' are simple graphs. Therefore, each edge in these graphs can be expressed as a 2-element set of vertices.

Let $f_1 : V \mapsto V'$ be the map defined by

$$f_1(u_i) = v_i$$

for each $i \in \{1, 2, \ldots, 10\}$. Let $f_2 : E \mapsto E'$ be the map defined by

$$f_2(\{u_i, u_j\}) = \{v_i, v_j\} \tag{2.2}$$

for every edge $\{u_i, u_j\}$ in E. The condition in Equation 2.2 is necessary for $f = (f_1, f_2)$ to be a homomorphism. The reader should verify that f is indeed a homomorphism.

Note that f_1 is a bijective map. The map $f_2 : E \mapsto E'$ is an injective map from a set E of 15 edges to another set E' of 15 edges. Therefore, f_2 is also a bijection. So, $f_1 : V \mapsto V'$ and $f_2 : E \mapsto E'$ are both bijective maps. Thus f is an isomorphism. This means that the graphs in Figure 2.7 are "equivalent"

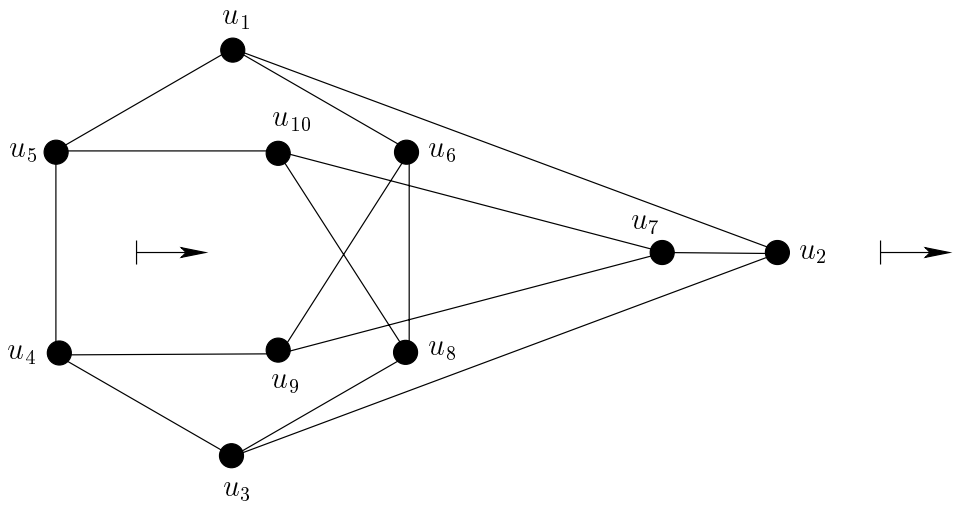


Figure 2.8: We stretch G to the right as indicated by the arrows in a step to deform G into G'.

in the sense that every graph-theoretic property fulfilled by one of the graphs is also fulfilled by the other.

In this example it is not hard to see that geometrically G' is a simple deformation of G. Hence, the isomorphism should come as no surprise. To see G' is a deformation of G simply take G and deform it by stretching to the right as illustrated in Figure 2.8. Next reflect the vertices u_2 and u_7 about the edge $\{u_6, u_8\}$ and let the edges with endvertices u_2 and u_7 follow. This is shown in Figure 2.9. We now see that G and G' "are the same."

The following theorem describes an equivalence relation among graphs.

Theorem 2.19
The isomorphism relation between graphs in a fixed collection is an equivalence relation. That is,

1. *For any graph G, $G \cong G$. In fact,*

$$\mathbf{I}_G = (\mathbf{I}_V, \mathbf{I}_E)$$

is always an automorphism of G.

2. *If $G \cong G'$ and $G' \cong G''$, then $G \cong G''$. In fact, if $f = (f_1, f_2)$ is an isomorphism from $G = (V, E, \phi)$ to $G' = (V', E', \phi')$ and $f' = (f'_1, f'_2)$ is an isomorphism from $G' = (V', E', \phi')$ to $G'' = (V'', E'', \phi'')$, then the ordered pair*

$$f' \circ f = (f'_1 \circ f_1, f'_2 \circ f_2)$$

is an isomorphism from G to G''.

3. *If $G \cong G'$ then $G' \cong G$. In fact, if $f = (f_1, f_2)$ is an isomorphism from $G = (V, E, \phi)$ to $G' = (V', E', \phi')$, then the ordered pair*

$$f^{-1} = (f_1^{-1}, f_2^{-1})$$

is an isomorphism from G' to G.

PROOF: Since the proofs of each of the three statements are very similar, we leave the proofs of the first two enumerated statements as exercises.

To prove the third statement, we assume that $f = (f_1, f_2)$ is an isomorphism from G to G'. Since both f_1 and f_2 are bijective, then so are their inverses f_1^{-1} and f_2^{-1}. To show $f^{-1} = (f_1^{-1}, f_2^{-1})$ is an isomorphism from G' to G, we need that the condition

$$\text{if } \phi'(e') = \{u', v'\} \text{ then } \phi(f_2^{-1}(e')) = \{f_1^{-1}(u'), f_1^{-1}(v')\}$$

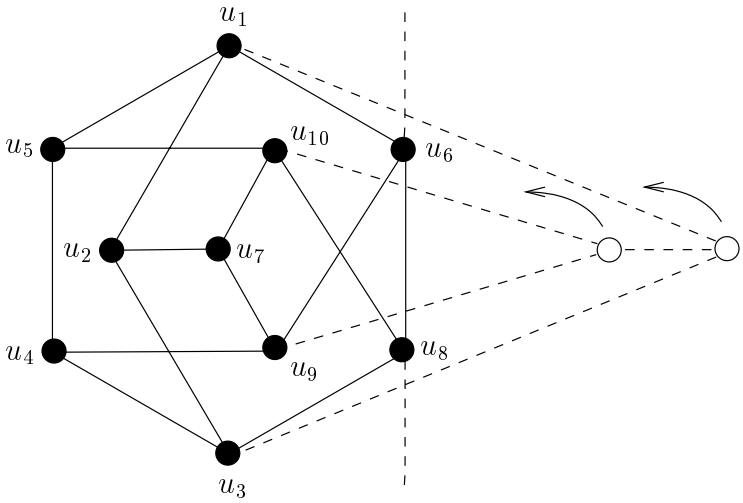


Figure 2.9: Reflection about the edge $\{u_6, u_8\}$ to obtain G' from G.

holds for every $e' \in E'$ and its endvertices $u', v' \in V'$.

Assume that $\phi'(e') = \{u', v'\}$ and $\phi(f_2^{-1}(e')) = \{x, y\}$. We want to show that $\{x, y\} = \{f_1^{-1}(u'), f_1^{-1}(v')\}$. Set $f_2^{-1}(e') = e$ so we have that $f_2(e) = e'$ and $\phi(e) = \{x, y\}$. Since $f = (f_1, f_2)$ is an isomorphism, we have

$$\{u', v'\} = \phi'(e') = \phi'(f_2(e)) = \{f_1(x), f_1(y)\}.$$

Therefore, either $u' = f_1(x)$ or $u' = f_1(y)$. For symmetric reasons we can assume the former. We then have that $u' = f_1(x)$ and $v' = f_1(y)$ (note this holds regardless of whether u' and v' are distinct). From this we get $f_1^{-1}(u') = x$ and $f_1^{-1}(v') = y$. Therefore, $\{x, y\} = \{f_1^{-1}(u'), f_1^{-1}(v')\}$. This proves $f^{-1} = (f_1^{-1}, f_2^{-1})$ is an isomorphism. □

Theorem 2.19 ensures that when considering graph-theoretic properties it suffices to study one particular graph of the equivalence class defined by $\cong$ (is isomorphic to). This is because any graph isomorphic to our chosen one would have the same graph-theoretic properties like "containing a vertex of degree greater than or equal to 23," "containing a simple path with more than half the number of vertices of the entire graph," "being connected," and so on. This is true because we can transform back and forth between the graphs in the equivalence class via the isomorphism.

This allows us to talk about "the graph" N_n (as we have done already, without justification, in Example 1.16 on page 16, and the examples following that one!) as just one representative of the countless graphs isomorphic to N_n. This is because all graphs isomorphic to N_n have exactly the same graph-theoretic properties as N_n. Therefore, they can be viewed to be the same graph as the graph represented in Example 1.16.

As we saw in Example 2.18 on page 50, the graphs G and G' represented by the diagrams in Figure 2.7 where isomorphic. Hence, we can say that G and G' are the same graph. This graph is called the *Petersen graph*. It is usually presented as the diagram of G on the left of Figure 2.7.

In the next section we look at homomorphisms and isomorphisms of simple graphs. By restricting ourselves to simple graphs, working with the corresponding definitions will be easier.

2.6 Homomorphisms and Isomorphisms of Simple Graphs

Notice that in Example 2.18 on page 50, the isomorphism from G to G' was completely determined by the vertex map f_1. This is the case for all simple graphs. Since most graph-theoretic problems can actually be reduced to the study of simple graphs, we can simplify our general definition (Definition 2.15 on page 47) of homomorphism and isomorphism between simple graphs. Recall that by Definition 1.15 on page 16, an edge in a simple graph is a two element subset of the set of its vertices.

Definition 2.20
Let $G = (V, E)$ and $G' = (V', E')$ be two simple graphs. A homomorphism

$$f : G \mapsto G'$$

from G to G' is an ordered pair $f = (f_1, f_2)$ of maps, $f_1 : V \mapsto V'$ and $f_2 : E \mapsto E'$ satisfying the compatibility condition

$$f_2(\{u, v\}) = \{f_1(u), f_1(v)\} \tag{2.3}$$

for every edge $\{u, v\} \in E$.

The homomorphism f is an isomorphism *if both f_1 and f_2 are bijective.*

Note that this definition is completely compatible with the more general Definition 2.15 on page 47 restricted to simple graphs. Hence, the isomorphism relation defines an equivalence relation between any collection of simple graphs as well.

Any subgraph of a given graph G, which is isomorphic to a complete graph K_l for some l, is called an *l-clique* in G. When the number of vertices in the subgraph is irrelevant, we simply call it a *clique*. With this definition in mind, the following items are worth noticing for homomorphisms between simple graphs:

1. Two adjacent vertices must be mapped to *different* adjacent vertices. So in particular, a clique is always mapped to a clique having the same number of vertices.

2. Suppose that $f = (f_1, f_2)$ is a homomorphism from G to G'. The vertex map $f_1 : V \mapsto V'$ completely determines the edge map $f_2 : E \mapsto E'$ by the compatibility condition (Equation 2.3).

Note that these two facts do not hold for graphs with loops and multiple edges.

Determining if two graphs are isomorphic is a well-known, but difficult open problem in graph theory. Given two graphs, the problem boils down to determining whether all graph-theoretic properties satisfied by one graph are also satisfied by the other. If we can spot a property that one graph satisfies but the other one does not, then we can rest assured that the graphs are not isomorphic. But even if two graphs are not isomorphic, finding a particular property satisfied by one and not the other can be quite tricky task even though we know for a fact that such a property exists.

There are times when we can tell right away that two graphs are not isomorphic. Although we discuss isomorphism between graphs further in Chapters 6 and 15, we now note a couple of properties that can easily be checked. Such properties are called *invariants* of a graph. If two graphs are isomorphic, they must satisfy the following:

1. They must have the same number of vertices and edges.
2. Their number of components must be the same.
3. For each $i \in \{0, 1, 2, \ldots\}$ the number of vertices with degree i must be the same in both graphs.

It becomes harder to determine whether two graphs are isomorphic when the number of edges is increased. When dealing only with simple graphs, we have a slight simplification. Recall that a simple graph with n vertices can have at most $(n-1)n/2$ edges. Theorem 2.19 shows that when considering two graphs both having n vertices and e edges we can restrict our attention to graphs with n vertices and at most $\lfloor (n-1)n/4 \rfloor$ edges. Before presenting that result, we need to introduce the notion of the *complement* of a graph.

Definition 2.21
For a simple graph $G = (V, E)$, the complement *of G is defined as*

$$\overline{G} = (V, \overline{E}),$$

where $\overline{E} = \{\{u, v\} : \{u, v\} \notin E\}$. In other words, $\overline{G}$ is a simple graph with the same set of vertices as G and where two vertices are adjacent if and only if they are not adjacent in G.

Note that $\overline{N_n}$ equals K_n and $\overline{K_n}$ equals N_n. In general, we have $\overline{\overline{G}}$ equals G for any simple graph G. With Definition 2.21 in mind, we obtain the following theorem.

Theorem 2.22
For simple graphs G and G', $G \cong G'$ if and only if $\overline{G} \cong \overline{G'}$.

PROOF: Exercise 21. □

Two simple graphs G and G' on n vertices are isomorphic if and only if their complements are isomorphic. So, we can conclude that if they both have e greater than $\lfloor (n-1)n/4 \rfloor$ edges, then the number of edges in their complements, e', is bounded above.

$$\begin{aligned} e' &= \frac{(n-1)n}{2} - e \\ &\leq \frac{(n-1)n}{2} - \left\lfloor \frac{(n-1)n}{4} \right\rfloor - 1 \\ &\leq \left\lfloor \frac{(n-1)n}{4} \right\rfloor \end{aligned}$$

In this case it might be easier to determine if the graphs are isomorphic by working with their complements.

In the following examples we consider several additional invariants that may be used to determine whether or not two graphs are isomorphic.

☞ **Example 2.23**
Consider the two graphs depicted in Figure 2.10.

Notice that the graph G on the left has two paths of length four. The graph G' on the right only has one path of length four. Therefore, G and G' cannot be isomorphic.

A more analytical argument works as follows: Both graphs G and G' are simple with six vertices and five edges. Both graphs have three vertices of

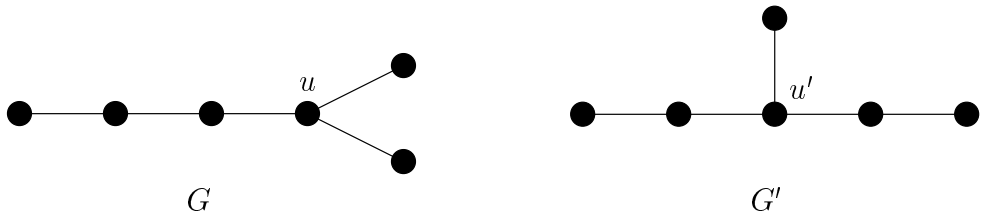


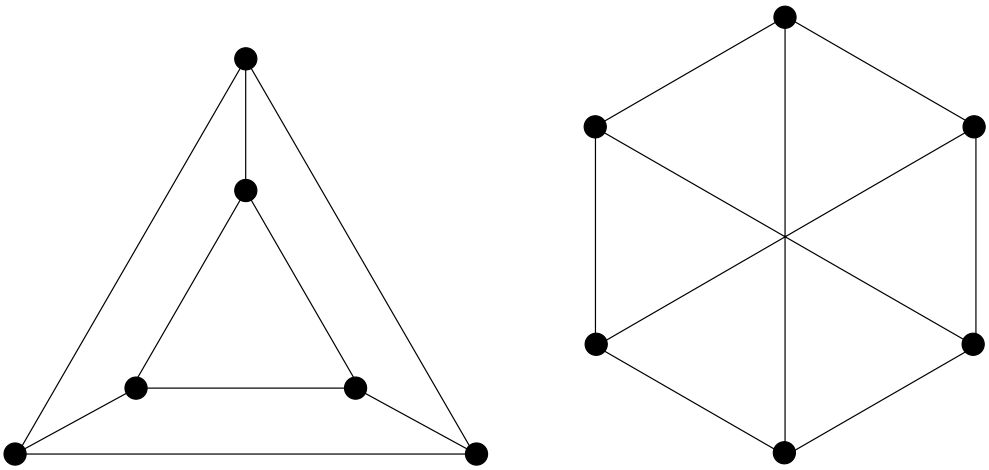
Figure 2.10: Graphs used in Example 2.23. Are the graphs isomorphic?

degree one, two vertices of degree two, and one vertex of degree three. Hence, if G and G' are isomorphic and $f : G \mapsto G'$ is an isomorphism, then $f_1(u) = u'$ must hold. This is because a vertex of degree k must be mapped to a vertex having degree k. A vertex adjacent to u must be mapped to a vertex adjacent to u'. Notice that the three vertices adjacent to u in G, two with degree one and one with degree two, must be mapped to the three vertices adjacent to u' in G', two of which have degree two and one which has degree one. This is clearly impossible, since as previously stated, a degree k vertex must be mapped to a degree k vertex. Thus we conclude that the two graphs G and G' are not isomorphic.

☞ **Example 2.24**
Consider the two graphs illustrated in Figure 2.11.

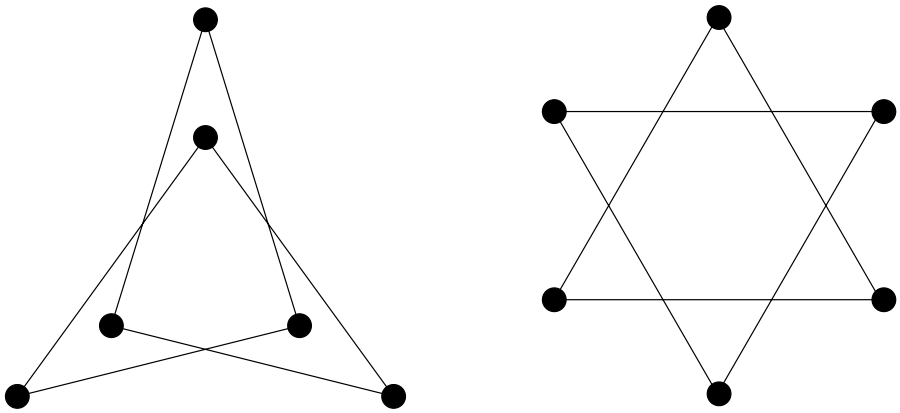
Both graphs are 3-regular simple graphs on six vertices. We note that the graph G on the left has a cycle of length three, but G' on the right has no 3-cycles. Therefore, these graphs are not isomorphic. Another argument can be achieved by examining the complements. The complements of the graphs are shown in Figure 2.12. From Figure 2.12 we see that the complement $\overline{G}$ is the cycle C_6 on six vertices. It is connected. However, the complement $\overline{G'}$ is the disjoint union of two copies of the cycle C_3 on three vertices. This graph has two components. So, $\overline{G}$ is not isomorphic to $\overline{G'}$. Hence, by Theorem 2.22, the graphs G and G' are not isomorphic either.

It is not hard to see that the graph G' on the right side of Figure 2.11 is isomorphic to $K_{3,3}$. As discussed in the Utilities Problem on page 5, $K_{3,3}$ is *nonplanar* in the sense that it cannot be drawn in the plane without edges crossing each other. Clearly, the graph G on the left side of Figure 2.10 is *planar*; there are no edge crossings. Hence, we have yet another way to see that G and G' are not isomorphic. We could continue to list graph-theoretic



G G'

Figure 2.11: Graphs used in Example 2.24. Are the graphs isomorphic?



$\overline{G} = C_6$ $\overline{G'} =$ "$2 \times C_3$"

Figure 2.12: Complements of the graphs shown in Figure 2.11.

properties satisfied by one graph and not by the other. However, just one such property suffices for the two graphs not to be isomorphic.

2.7 Exercises

1. Is it possible to draw a graph that has a trail of length seven but no path of length seven? If so, draw such an example. Is it possible to draw a graph that has a path of length seven but no trail of length seven? If so, draw such an example.

2. Represent the maze shown in Figure 2.13 with a graph such that a vertex denotes either a corridor or a dead end (as numbered). An edge represents a possible path between two vertices. (This is one of numerous mazes that were drawn or built by the Arabs, Greeks, Hindus, Romans, Vikings, and others. See also Exercise 3.)

3. What is the length of the path from the entrance to the center of the maze shown in Exercise 2?

4. When considered separately, for what values of n do N_n, P_n, C_n, K_n, and $K_{n,n}$ have

   (a) trails of length n

   (b) walks of length n

   (c) cycles of length n

5. Show that if a graph G contains a closed trail of odd length, then G contains a cycle of odd length. Does the same hold if we replace the word "odd" with "even" throughout the previous sentence?

6. Prove that a simple graph with n vertices must be connected if it has more than
$$\frac{(n-1)(n-2)}{2}$$
edges. [*Hint:* Use Theorem 2.14.]

7. Prove that a connected graph G remains connected after removing an edge e if and only if e is in some circuit of G.

8. Draw a connected graph with at least six vertices that becomes disconnected when any edge is removed.

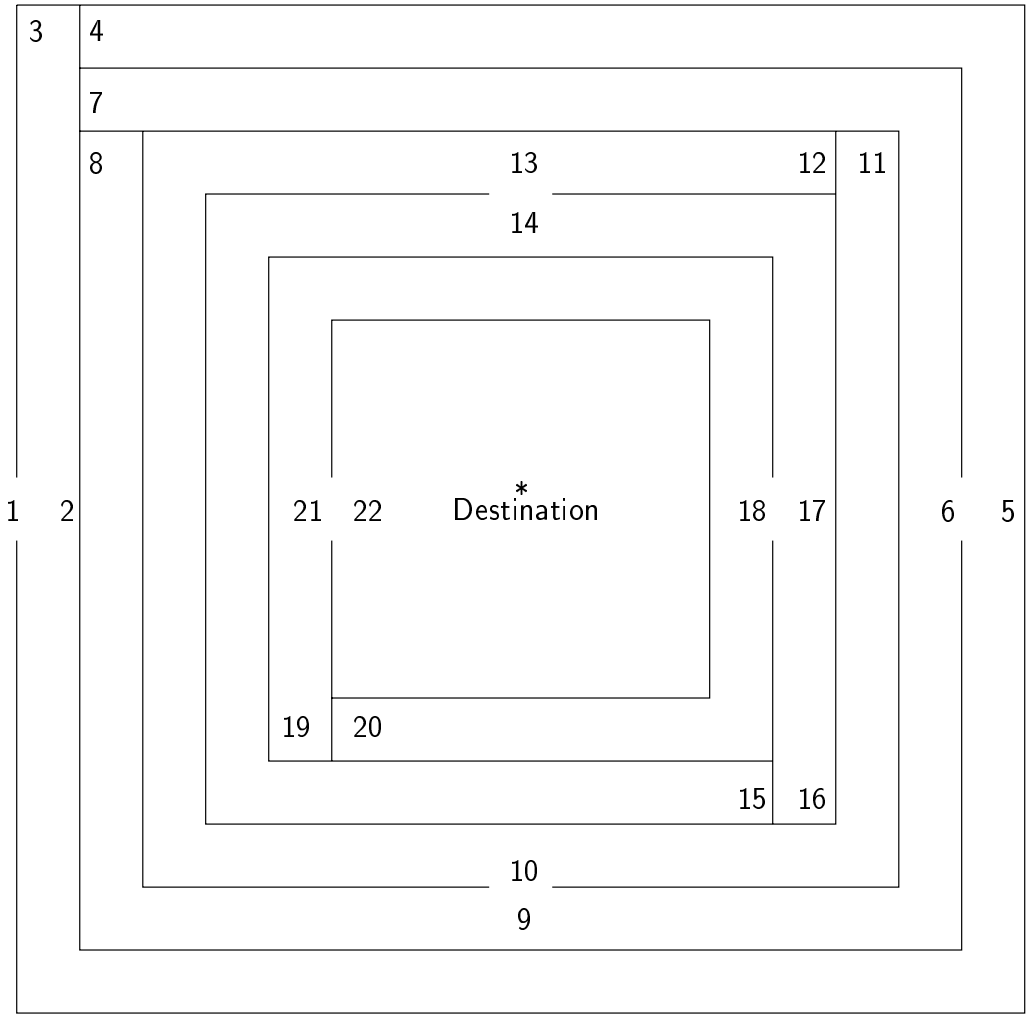


Figure 2.13: A maze.

9. Prove that a graph with n vertices satisfying the conditions of Exercise 8 is simple and has exactly $n - 1$ edges.

10. Define a relation on graphs such that G and G' are related if and only if the degree of the maximum degree vertex of G is less than or equal to the degree of the maximum degree vertex of G'? Does this relationship satisfy the reflexive, transitive, and antisymmetric properties? Is it a partial order? Does it satisfy the symmetric property? Is the relation an equivalence

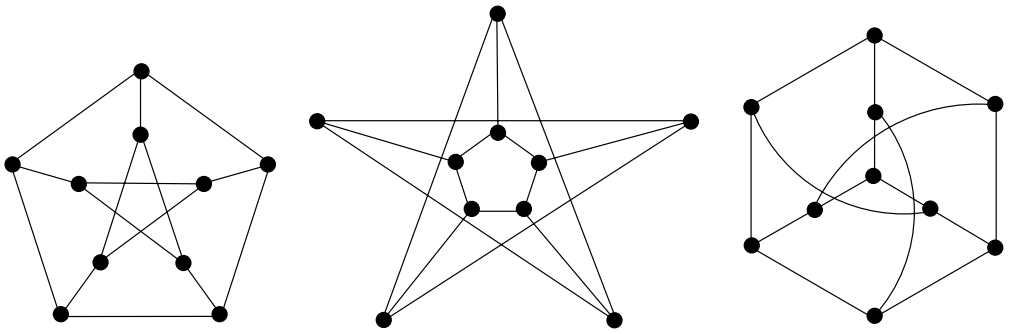


Figure 2.14: Three isomorphic graphs used in Exercise 14.

relation?

11. A connected graph is called 2-*connected* if the deletion of any single vertex does not split the graph into two components. For which, if any values of n are N_n, P_n, C_n, K_n, and $K_{n,n}$ 2-connected?

12. For each natural number n greater than or equal to three, what is the maximum number of edges a simple graph with n vertices can have and contain three components? For n equal to seven draw a graph that realizes this maximum? What are the degrees of its vertices?

13. Suppose G and G' are two graphs having n vertices. For what values of n is it possible for G to have more components and edges than G'?

14. Verify that the three graphs shown in Figure 2.14 are isomorphic. Label the corresponding vertices and edges.

15. Show that the two graphs depicted in Figure 2.15 are isomorphic.

16. Enter a query of "Graph Isomorphism" to your favorite search engine. Read a number of the Web pages selecting from the hyperlinks returned by your query. Summarize your findings in a one page paper.

17. Construct three examples to show that conditions 1, 2, and 3 on page 55 are not sufficient for isomorphism between graphs.

18. Show that for n greater than one, the graph $K_{n,2n}$ has a subgraph isomorphic to C_k for any even $4 \leq k \leq 2n$.

19. Prove that any two simple connected graphs with n vertices, all having degree two, are isomorphic.

20. Are the two graphs shown in Figure 2.16 isomorphic? If so, provide an isomorphism between them. If not, provide a clear explanation why not.

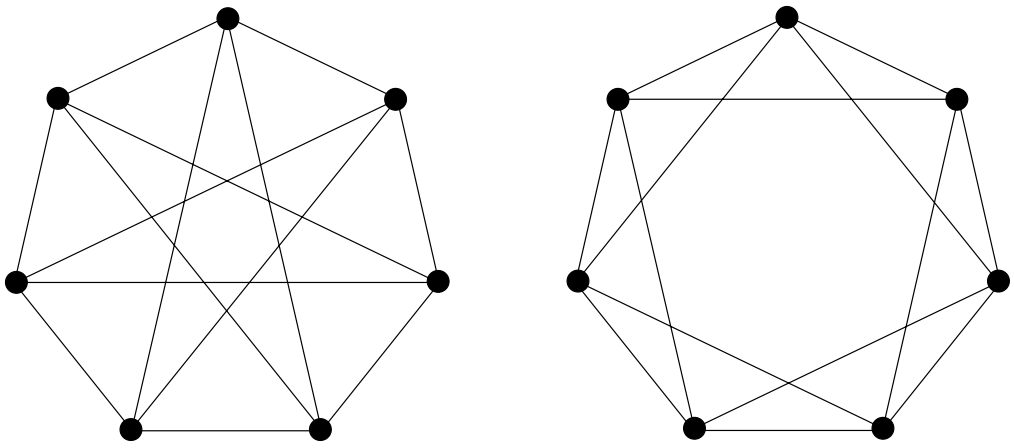
21. Prove Theorem 2.22. [*Hint:* Use one isomorphism to establish the other isomorphism and use the fact that $\overline{\overline{G}}$ equals G.]

22. Are any of the graphs N_n, P_n, C_n, K_n, and $K_{n,n}$ complements of each other?

23. Prove whether or not the complement of every regular graph is regular.

24. Show that if a simple graph G is isomorphic to its complement $\overline{G}$, then G has either $4k$ or $4k+1$ vertices for some natural number k. Find all simple graphs on four and five vertices that are isomorphic to their complements.

25. Let G be a simple graph. Show that either G or its complement $\overline{G}$ is connected.



(a) (b)

Figure 2.15: Isomorphic graphs used in Exercise 15.

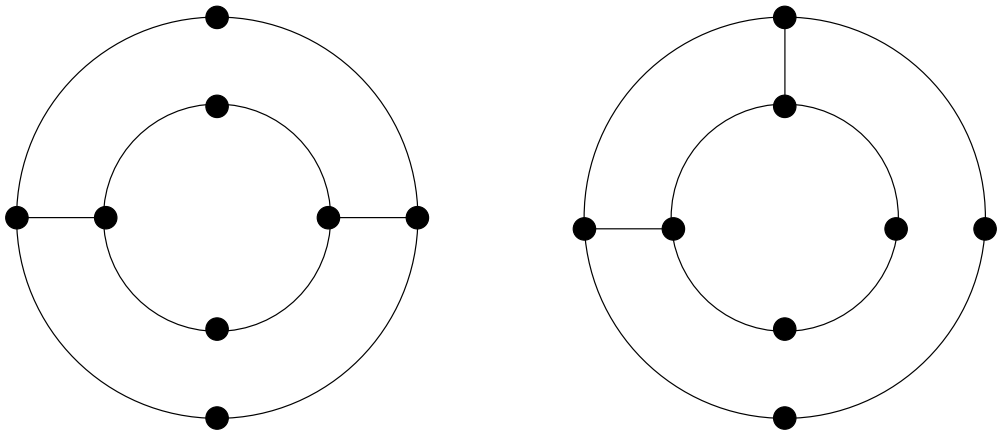


Figure 2.16: Graphs used in Exercise 20. Are these graphs isomorphic?

26. A newspaper reporter from the New York Times has contacted you to write a one page article for the Times that makes the concepts of connectedness and isomorphism understandable to the masses. Develop such an article being sure to include lots of intuition about these concepts.