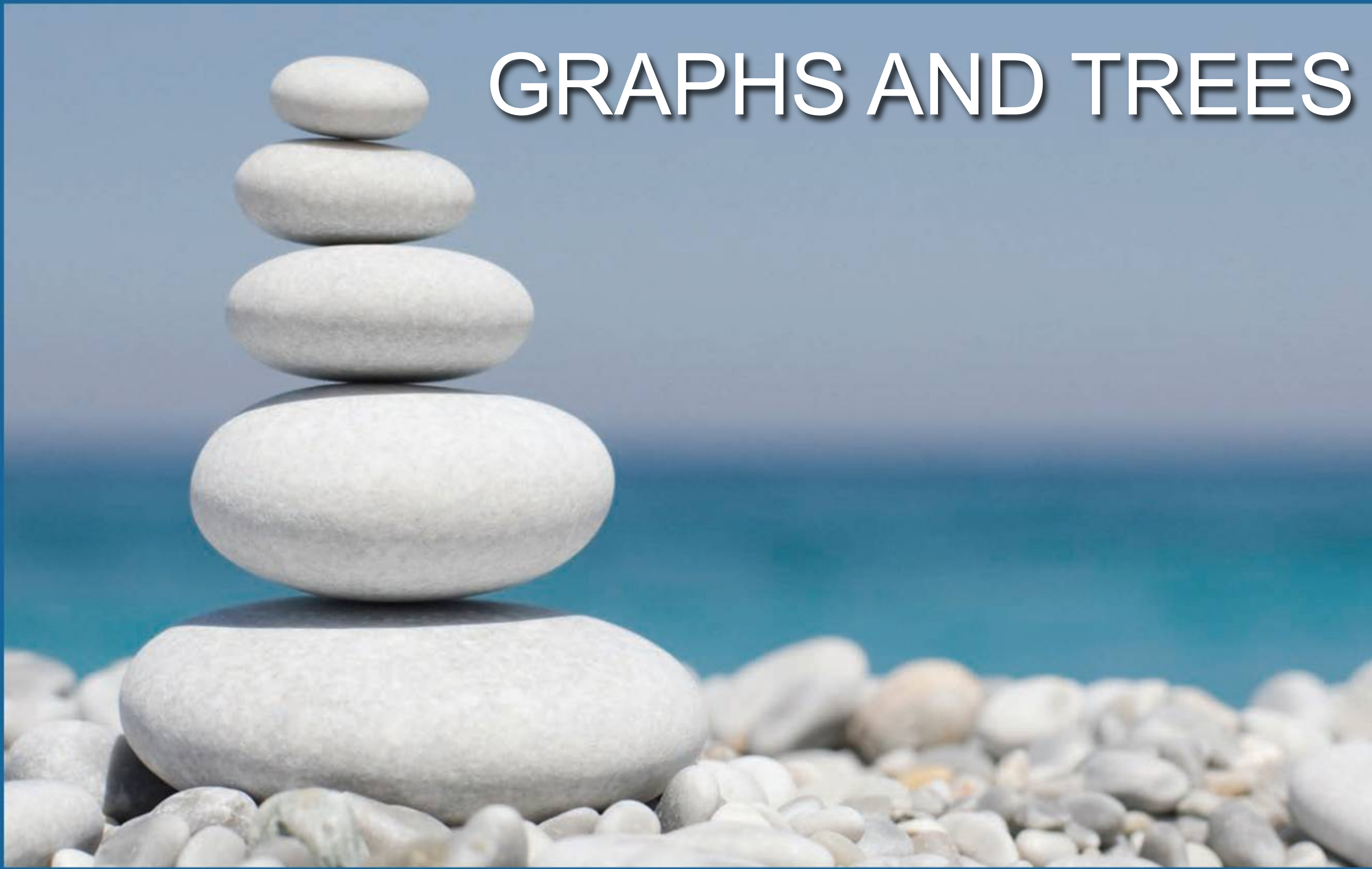
CHAPTER 10

GRAPHS AND TREES



Alessandro Artale – UniBZ - http://www.inf.unibz.it/~artale/z

SECTION 10.1

Graphs: Definitions and Basic Properties

Copyright © Cengage Learning. All rights reserved.

Graphs: Definitions and Basic Properties

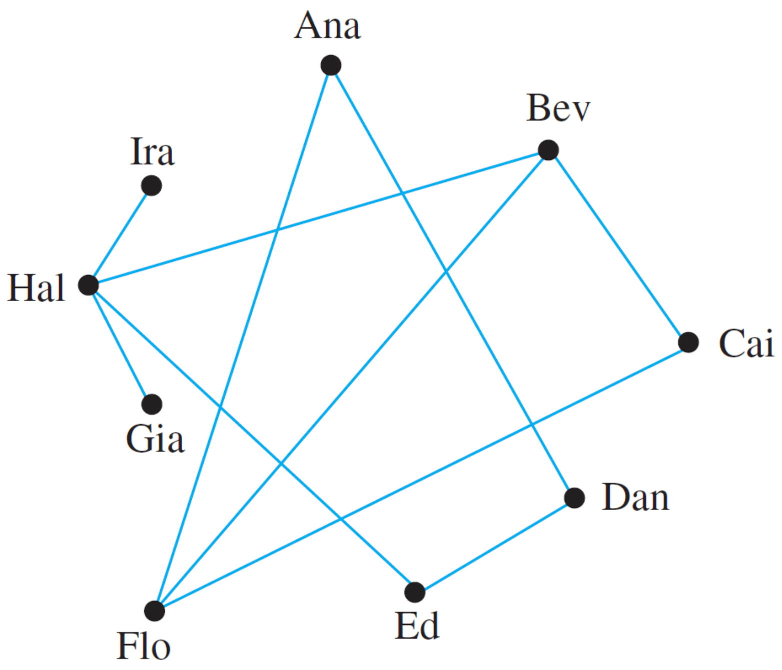
Imagine an organization that wants to set up teams of three to work on some projects.

In order to maximize the number of people on each team who had previous experience working together successfully, the director asked the members to provide names of their past partners.

Graphs: Definitions and Basic Properties

This information is displayed below both in a table and in a diagram.

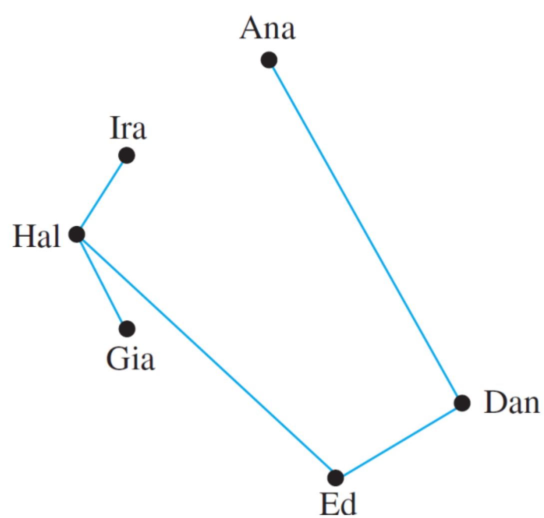
Name	Past Partners
Ana	Dan, Flo
Bev	Cai, Flo, Hal
Cai	Bev, Flo
Dan	Ana, Ed
Ed	Dan, Hal
Flo	Cai, Bev, Ana
Gia	Hal
Hal	Gia, Ed, Bev, Ira
Ira	Hal



Graphs: Definitions and Basic Properties

From the diagram, it is easy to see that Bev, Cai, and Flo are a group of three past partners, and so they should form one of these teams.

The following figure shows the result when these three names are removed from the diagram.



Graphs: Definitions and Basic Properties

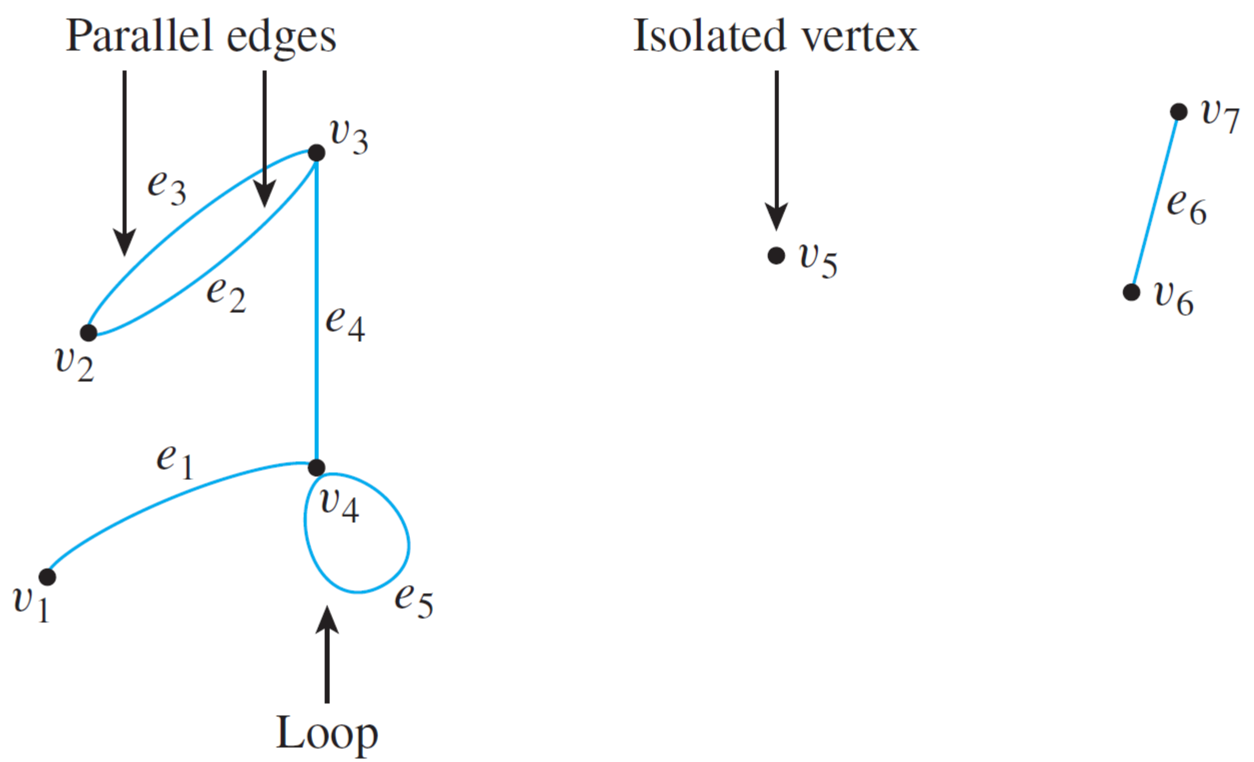
This drawing shows that placing Hal on the same team as Ed would leave Gia and Ira on a team containing no past partners.

However, if Hal is placed on a team with Gia and Ira, then the remaining team would consist of Ana, Dan, and Ed, and both teams would contain at least one pair of past partners.

Such drawings are illustrations of a structure known as a ***graph***. The dots are called *vertices* and the line segments joining vertices are called *edges*.

Graphs: Definitions and Basic Properties

The edges may be straight or curved and should either connect one vertex to another or a vertex to itself, as shown below.



Graphs: Definitions and Basic Properties

In this drawing, the vertices have been labeled with *v*' s and the edges with *e*' s.

When an edge connects a vertex to itself (as e_5 does), it is called a *loop*. When two edges connect the same pair of vertices (as e_2 and e_3 do), they are said to be *parallel*.

It is quite possible for a vertex to be unconnected by an edge to any other vertex in the graph (as v_5 is), and in that case the vertex is said to be *isolated*.

Graphs: Definitions and Basic Properties

The formal definition of a graph follows.

• Definition

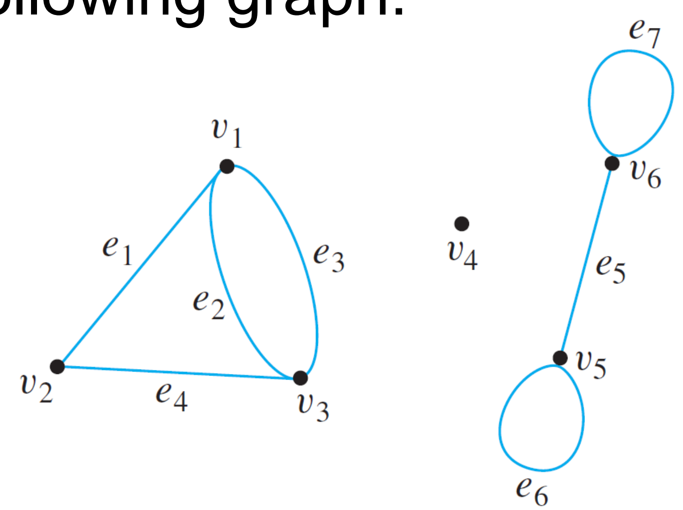
A **graph** G consists of two finite sets: a nonempty set $V(G)$ of **vertices** and a set $E(G)$ of **edges,** where each edge is associated with a set consisting of either one or two vertices called its **endpoints.** The correspondence from edges to endpoints is called the **edge-endpoint function.**

An edge with just one endpoint is called a **loop,** and two or more distinct edges with the same set of endpoints are said to be **parallel.** An edge is said to **connect** its endpoints; two vertices that are connected by an edge are called **adjacent;** and a vertex that is an endpoint of a loop is said to be **adjacent to itself.**

An edge is said to be **incident on** each of its endpoints, and two edges incident on the same endpoint are called **adjacent.** A vertex on which no edges are incident is called **isolated.**

Example 1 – *Terminology*

Consider the following graph:



a. Write the vertex set and the edge set, and give a table showing the edge-endpoint function.

b. Find all edges that are incident on v_1, all vertices that are adjacent to v_1, all edges that are adjacent to e_1, all loops, all parallel edges, all vertices that are adjacent to themselves, and all isolated vertices.

Example 1(a) – *Solution*

vertex set = $\{v_1, v_2, v_3, v_4, v_5, v_6\}$
edge set = $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$

edge-endpoint function:

Edge	Endpoints
e_1	$\{v_1, v_2\}$
e_2	$\{v_1, v_3\}$
e_3	$\{v_1, v_3\}$
e_4	$\{v_2, v_3\}$
e_5	$\{v_5, v_6\}$
e_6	$\{v_5\}$
e_7	$\{v_6\}$

Example 1(a) – *Solution*

cont'd

Note that the isolated vertex v_4 does not appear in this table.

Although each edge must have either one or two endpoints, a vertex need not be an endpoint of an edge.

Example 1(b) – *Solution*

cont'd

e_1, e_2, and e_3 are incident on v_1.

v_2 and v_3 are adjacent to v_1.

e_2, e_3, and e_4 are adjacent to e_1.

e_6 and e_7 are loops.

e_2 and e_3 are parallel.

v_5 and v_6 are adjacent to themselves.

v_4 is an isolated vertex.

Graphs: Definitions and Basic Properties

We have discussed the directed graph of a binary relation on a set.

The general definition of **directed graph** is similar to the definition of graph, except that one associates an *ordered pair* of vertices with each edge instead of a *set* of vertices.

Thus each edge of a directed graph can be drawn as an arrow going from the first vertex to the second vertex of the ordered pair.

Graphs: Definitions and Basic Properties

- **Definition**

A **directed graph, or digraph,** consists of two finite sets: a nonempty set $V(G)$ of vertices and a set $D(G)$ of directed edges, where each is associated with an ordered pair of vertices called its **endpoints.** If edge e is associated with the pair (v, w) of vertices, then e is said to be the (**directed**) **edge** from v to w.

Note that each directed graph has an associated ordinary (undirected) graph, which is obtained by ignoring the directions of the edges.

Examples of Graphs

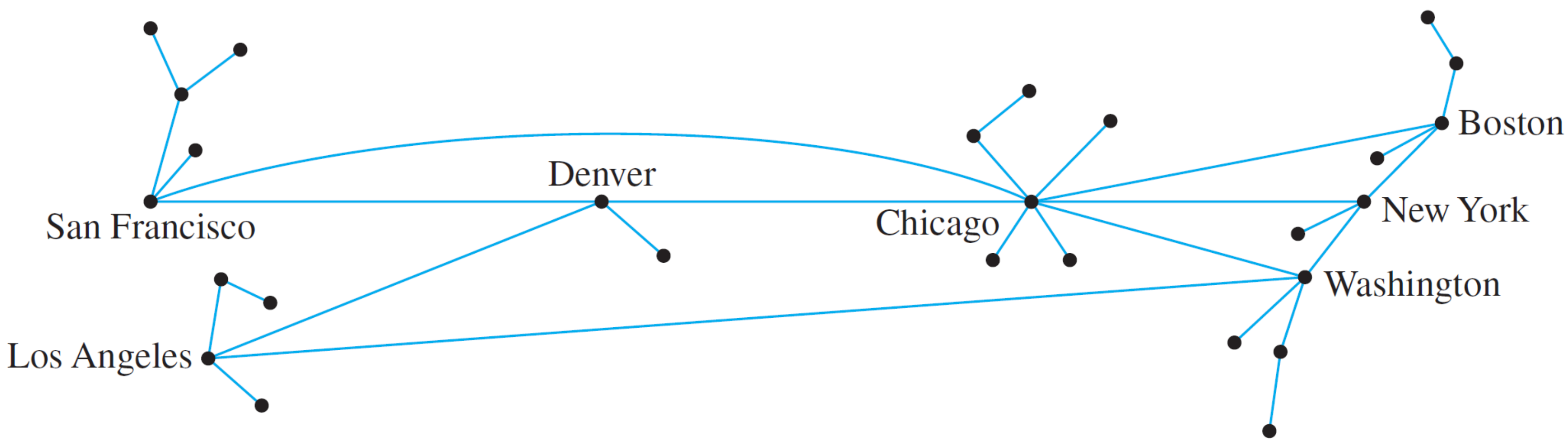
Example 4 – *Using a Graph to Represent a Network*

Telephone, electric power, gas pipeline, and air transport systems can all be represented by graphs, as can computer networks—from small local area networks to the global Internet system that connects millions of computers worldwide.

Questions that arise in the design of such systems involve choosing connecting edges to minimize cost, optimize a certain type of service, and so forth.

Example 4 – *Using a Graph to Represent a Network* cont'd

A typical network, called a hub and spoke model, is shown below.



Special Graphs

Special Graphs

One important class of graphs consists of those that do not have any loops or parallel edges.

Such graphs are called ***simple***. In a simple graph, no two edges share the same set of endpoints, so specifying two endpoints is sufficient to determine an edge.

• Definition and Notation

A **simple graph** is a graph that does not have any loops or parallel edges. In a simple graph, an edge with endpoints v and w is denoted $\{v, w\}$.

Example 8 – *A Simple Graph*

Draw all simple graphs with the four vertices $\{u, v, w, x\}$ and two edges, one of which is $\{u, v\}$.

Solution:

Each possible edge of a simple graph corresponds to a subset of two vertices.

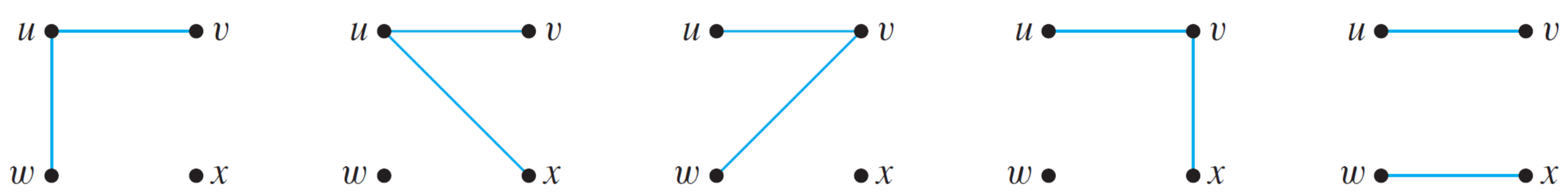
Given four vertices, there are $\binom{4}{2} = 6$ such subsets in all: $\{u, v\}$, $\{u, w\}$, $\{u, x\}$, $\{v, w\}$, $\{v, x\}$, and $\{w, x\}$.

Example 8 – *Solution*

cont' d

Now one edge of the graph is specified to be $\{u, v\}$, so any of the remaining five from this list can be chosen to be the second edge.

The possibilities are shown below.



Special Graphs

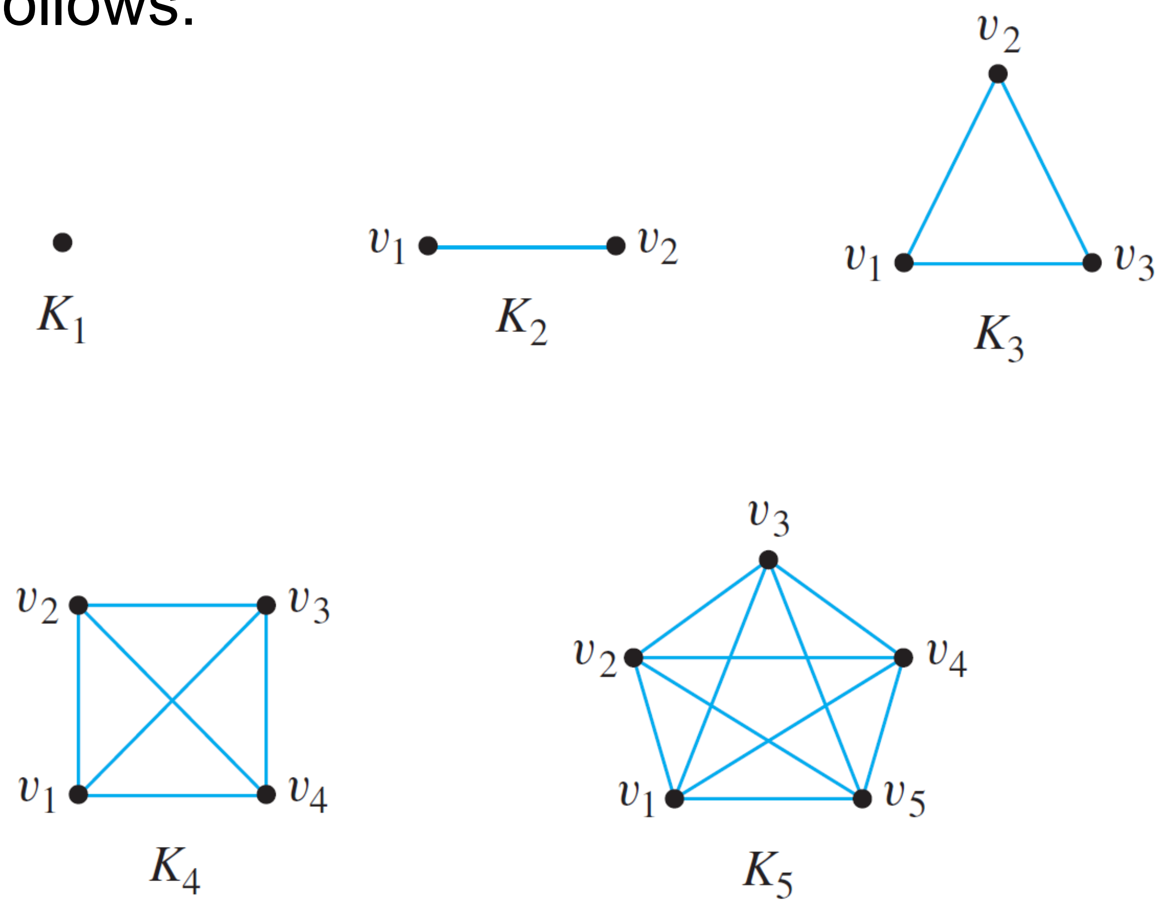
Another important class of graphs consists of those that are "**complete**" in the sense that all pairs of vertices are connected by edges.

• Definition

Let n be a positive integer. A **complete graph on *n* vertices,** denoted K_n, is a simple graph with n vertices and exactly one edge connecting each pair of distinct vertices.

Example 9 – *Complete Graphs on n Vertices: K_1, K_2, K_3, K_4, K_5*

The complete graphs K_1, K_2, K_3, K_4, and K_5 can be drawn as follows:



Special Graphs

In yet another class of graphs, the vertex set can be separated into two subsets: Each vertex in one of the subsets is connected by exactly one edge to each vertex in the other subset, but not to any vertex in its own subset. Such a graph is called ***complete bipartite***.

• Definition

Let m and n be positive integers. A **complete bipartite graph on (m, n) vertices,** denoted $\boldsymbol{K_{m,n}}$, is a simple graph with distinct vertices $v_1, v_2, \ldots, v_m$ and $w_1, w_2, \ldots, w_n$ that satisfies the following properties: For all $i, k = 1, 2, \ldots, m$ and for all $j, l = 1, 2, \ldots, n$,

1. There is an edge from each vertex v_i to each vertex w_j.
2. There is no edge from any vertex v_i to any other vertex v_k.
3. There is no edge from any vertex w_j to any other vertex w_l.

Special Graphs

- **Definition**

A graph H is said to be a **subgraph** of a graph G if, and only if, every vertex in H is also a vertex in G, every edge in H is also an edge in G, and every edge in H has the same endpoints as it has in G.

The Concept of Degree

The Concept of Degree

The ***degree of a vertex*** is the number of end segments of edges that "stick out of" the vertex.

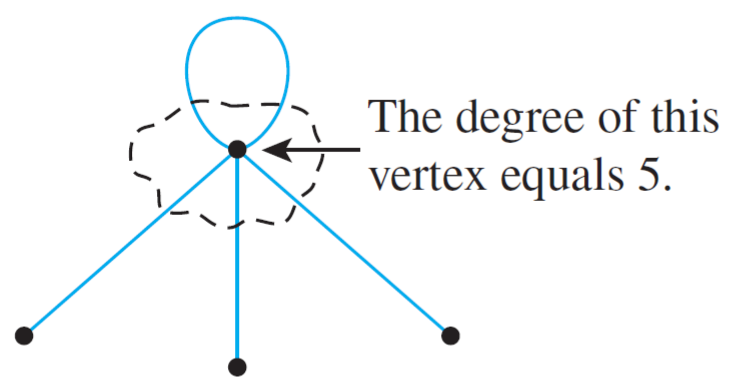
• Definition

Let G be a graph and v a vertex of G. The **degree of v,** denoted **deg(v),** equals the number of edges that are incident on v, with an edge that is a loop counted twice. The **total degree of G** is the sum of the degrees of all the vertices of G.

The Concept of Degree

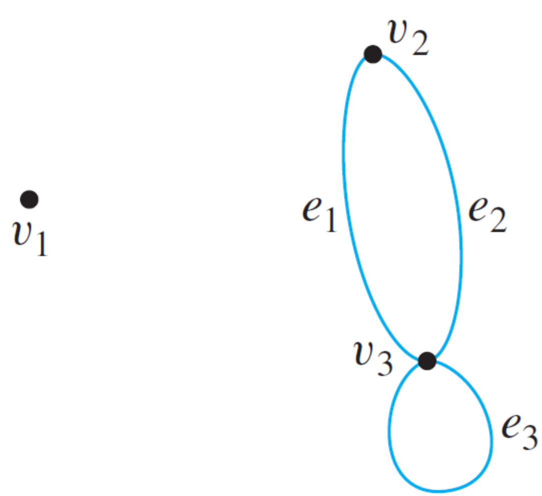
Since an edge that is a loop is counted twice, the degree of a vertex can be obtained from the drawing of a graph by counting how many end segments of edges are incident on the vertex.

This is illustrated below.



Example 12 – *Degree of a Vertex and Total Degree of a Graph*

Find the degree of each vertex of the graph *G* shown below. Then find the total degree of *G*.



Example 12 – *Solution*

deg(v_1) = 0 since no edge is incident on v_1 (v_1 is isolated).

deg(v_2) = 2 since both e_1 and e_2 are incident on v_2.

deg(v_3) = 4 since e_1 and e_2 are incident on v_3 and the loop e_3 is also incident on v_3 (and contributes 2 to the degree of v_3).

total degree of G = deg(v_1) + deg(v_2) + deg(v_3)

= 0 + 2 + 4

= 6.

The Concept of Degree

Note that the total degree of the graph G of Example 12, which is 6, equals twice the number of edges of G, which is 3.

Roughly speaking, this is true because each edge has two end segments, and each end segment is counted once toward the degree of some vertex. This result generalizes to any graph.

We will show that the sum of the degrees of all the vertices in a graph is twice the number of edges of the graph.

The Concept of Degree

For any graph without loops, the general result can be explained as follows: Imagine a group of people at a party where each person shakes hands with other people.

Each person participates in a certain number of handshakes—perhaps many, perhaps none—but because each handshake is experienced by two different people, if the numbers experienced by each person are added together, the sum will equal twice the total number of handshakes.

This is such an attractive way of understanding the situation that the following theorem is often called the *handshake lemma* or the *handshake theorem*.

The Concept of Degree

As the proof demonstrates, the conclusion is true even if the graph contains loops.

Theorem 10.1.1 The Handshake Theorem

If G is any graph, then the sum of the degrees of all the vertices of G equals twice the number of edges of G. Specifically, if the vertices of G are $v_1, v_2, \ldots, v_n$, where n is a nonnegative integer, then

$$\begin{aligned}\text{the total degree of } G &= \deg(v_1) + \deg(v_2) + \cdots + \deg(v_n)\\ &= 2 \cdot (\text{the number of edges of } G).\end{aligned}$$

Corollary 10.1.2

The total degree of a graph is even.

The Concept of Degree

The following proposition is easily deduced from Corollary 10.1.2 using properties of even and odd integers.

Proposition 10.1.3

In any graph there are an even number of vertices of odd degree.

SECTION 10.2

Trails, Paths, and Circuits

Copyright © Cengage Learning. All rights reserved.

Trails, Paths, and Circuits

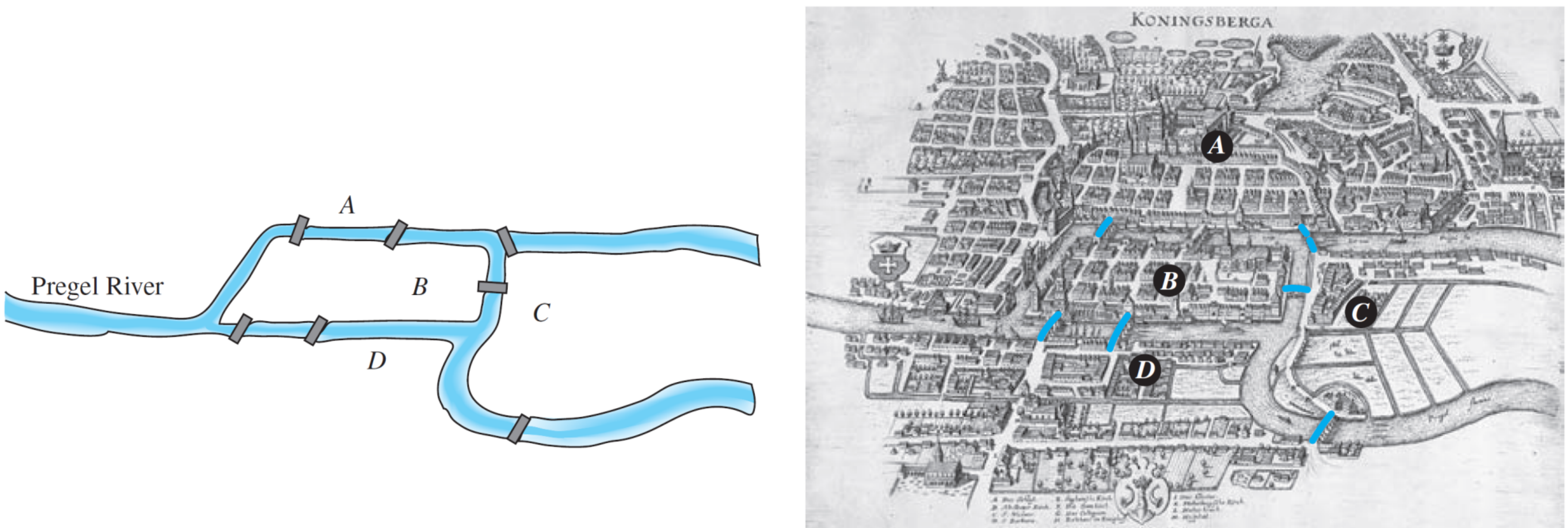
The subject of graph theory began in the year 1736 when the great mathematician Leonhard Euler published a paper giving the solution to the following puzzle:

The town of Königsberg in Prussia (now Kaliningrad in Russia) was built at a point where two branches of the Pregel River came together.

It consisted of an island and some land along the river banks.

Trails, Paths, and Circuits

These were connected by seven bridges as shown in Figure 10.2.1.



The Seven Bridges of Königsberg

Figure 10.2.1

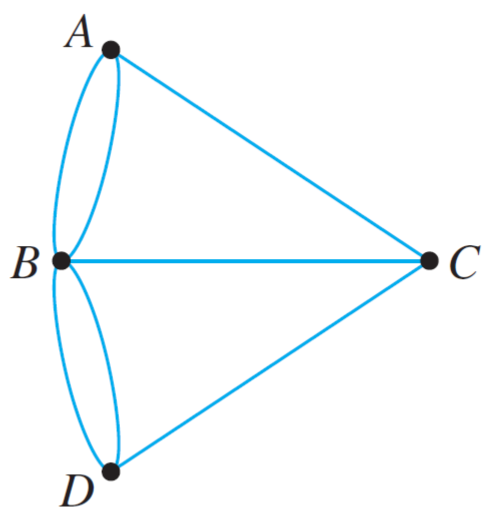
Trails, Paths, and Circuits

The question is this: Is it possible for a person to take a walk around town, starting and ending at the same location and crossing each of the seven bridges exactly once?

To solve this puzzle, Euler translated it into a graph theory problem. He noticed that all points of a given land mass can be identified with each other since a person can travel from any one point to any other point of the same land mass without crossing a bridge.

Trails, Paths, and Circuits

Thus for the purpose of solving the puzzle, the map of Königsberg can be identified with the graph shown in Figure 10.2.2, in which the vertices *A*, *B*, *C*, and *D* represent land masses and the seven edges represent the seven bridges.



Graph Version of Königsberg Map

Figure 10.2.2

Trails, Paths, and Circuits

In terms of this graph, the question becomes the following:

> Is it possible to find a route through the graph that starts and ends at some vertex, one of *A*, *B*, *C*, or *D*, and traverses each edge exactly once?

Equivalently:

> Is it possible to trace this graph, starting and ending at the same point, without ever lifting your pencil from the paper?

Trails, Paths, and Circuits

If you start at vertex *A*, for example, each time you pass through vertex *B* and *C* (or *D*) you use up two edges because you arrive on one edge and depart on a different one.

So, if it is possible to find a route that uses all the edges of the graph and starts and ends at *A*, then the total number of arrivals and departures from each vertex *B*, *C*, and *D* must be a multiple of 2: the degrees of the vertices *B*, *C*, and *D* must be even.

But they are not: deg(*B*) = 5, deg(*C*) = 3, and deg(*D*) = 3. Hence there is no route that solves the puzzle by starting and ending at *A*.

Trails, Paths, and Circuits

Similar reasoning can be used to show that there are no routes that solve the puzzle by starting and ending at *B*, *C*, or *D*.

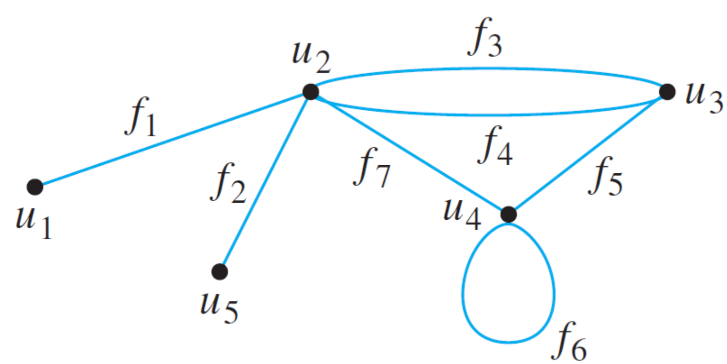
Therefore, it is impossible to travel all around the city crossing each bridge exactly once.

Definitions

Definitions

Travel in a graph is accomplished by moving from one vertex to another along a sequence of adjacent edges.

In the graph below, for instance, you can go from u_1 to u_4 by taking f_1 to u_2 and then f_7 to u_4. This is represented by writing $u_1f_1u_2f_7u_4$.



Definitions

Or you could take the roundabout route

$$u_1 f_1 u_2 f_3 u_3 f_4 u_2 f_3 u_3 f_5 u_4 f_6 u_4 f_7 u_2 f_3 u_3 f_5 u_4 .$$

Certain types of sequences of adjacent vertices and edges are of special importance in graph theory: those that do not have a repeated edge, those that do not have a repeated vertex, and those that start and end at the same vertex.

Definitions

• Definition

Let G be a graph, and let v and w be vertices in G.

A **walk from *v* to *w*** is a finite alternating sequence of adjacent vertices and edges of G. Thus a walk has the form

$$v_0 e_1 v_1 e_2 \cdots v_{n-1} e_n v_n,$$

where the v's represent vertices, the e's represent edges, $v_0 = v$, $v_n = w$, and for all $i = 1, 2, \ldots n$, v_{i-1} and v_i are the endpoints of e_i. The **trivial walk from *v* to *v*** consists of the single vertex v.

A **trail from *v* to *w*** is a walk from v to w that does not contain a repeated edge.

A **path from *v* to *w*** is a trail that does not contain a repeated vertex.

A **closed walk** is a walk that starts and ends at the same vertex.

A **circuit** is a closed walk that contains at least one edge and does not contain a repeated edge.

A **simple circuit** is a circuit that does not have any other repeated vertex except the first and last.

Definitions

For ease of reference, these definitions are summarized in the following table:

	Repeated Edge?	Repeated Vertex?	Starts and Ends at Same Point?	Must Contain at Least One Edge?
Walk	allowed	allowed	allowed	no
Trail	no	allowed	allowed	no
Path	no	no	no	no
Closed walk	allowed	allowed	yes	no
Circuit	no	allowed	yes	yes
Simple circuit	no	first and last only	yes	yes

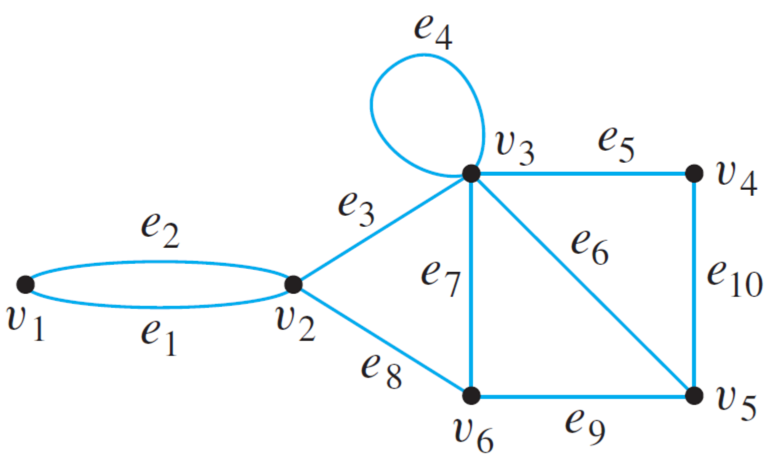
Often a walk can be specified unambiguously by giving either a sequence of edges or a sequence of vertices.

Example 2 – *Walks, Trails Paths, and Circuits*

In the graph below, determine which of the following walks are trails, paths, circuits, or simple circuits.

a. $v_1 e_1 v_2 e_3 v_3 e_4 v_3 e_5 v_4$ **b.** $e_1 e_3 e_5 e_5 e_6$ **c.** $v_2 v_3 v_4 v_5 v_3 v_6 v_2$

d. $v_2 v_3 v_4 v_5 v_6 v_2$ **e.** $v_1 e_1 v_2 e_1 v_1$ **f.** v_1



Example 2 – *Solution*

a. This walk has a repeated vertex but does not have a repeated edge, so it is a trail from v_1 to v_4 but not a path.

b. This is just a walk from v_1 to v_5. It is not a trail because it has a repeated edge.

c. This walk starts and ends at v_2, contains at least one edge, and does not have a repeated edge, so it is a circuit. Since the vertex v_3 is repeated in the middle, it is not a simple circuit.

d. This walk starts and ends at v_2, contains at least one edge, does not have a repeated edge, and does not have a repeated vertex. Thus it is a simple circuit.

Example 2 – *Solution*

cont'd

e. This is just a closed walk starting and ending at v_1. It is not a circuit because edge e_1 is repeated.

f. The first vertex of this walk is the same as its last vertex, but it does not contain an edge, and so it is not a circuit. It is a closed walk from v_1 to v_1. (It is also a trail from v_1 to v_1.)

Connectedness

Connectedness

It is easy to understand the concept of connectedness on an intuitive level.

Roughly speaking, a graph is **connected** if it is possible to travel from any vertex to any other vertex along a sequence of adjacent edges of the graph.

Connectedness

The formal definition of connectedness is stated in terms of walks.

• Definition

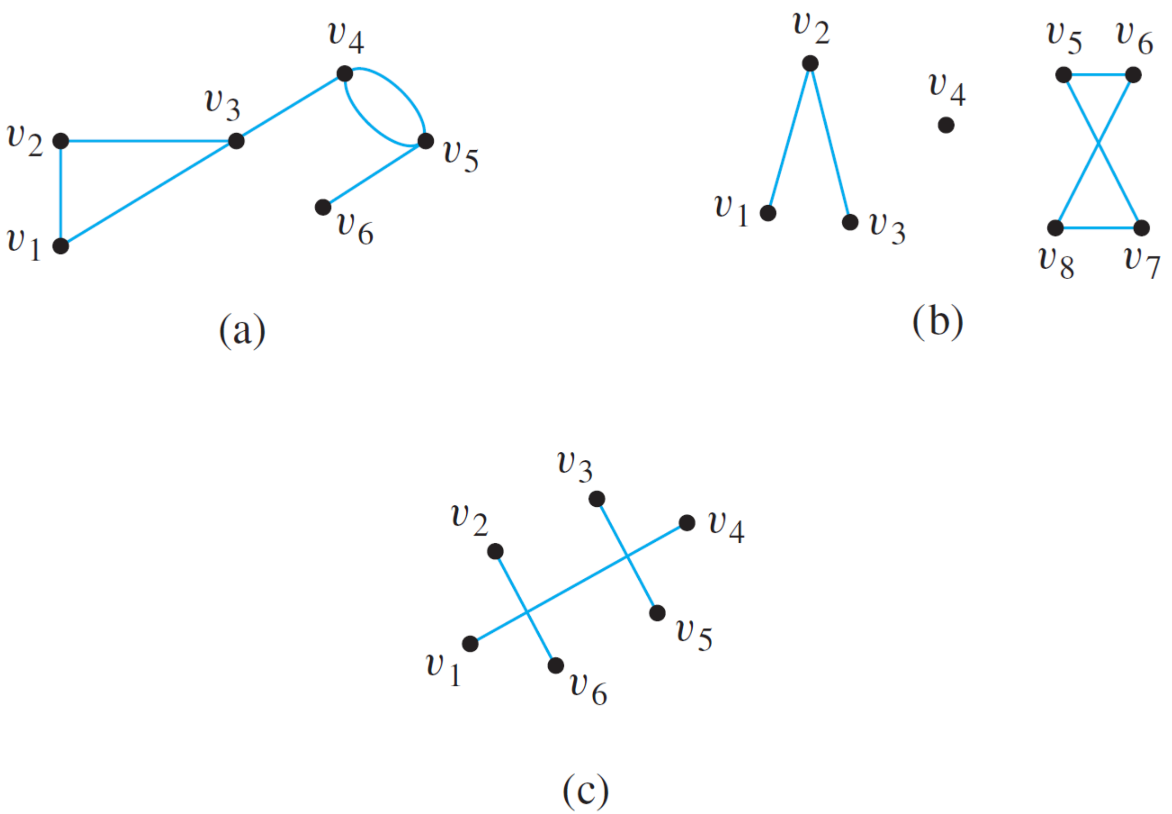
Let G be a graph. Two **vertices v and w of G are connected** if, and only if, there is a walk from v to w. The **graph G is connected** if, and only if, given *any* two vertices v and w in G, there is a walk from v to w. Symbolically,

$$G \text{ is connected} \quad \Leftrightarrow \quad \forall \text{ vertices } v, w \in V(G), \exists \text{ a walk from } v \text{ to } w.$$

If you take the negation of this definition, you will see that a graph G *is not connected* if, and only if, there are two vertices of G that are not connected by any walk.

Example 3 – *Connected and Disconnected Graphs*

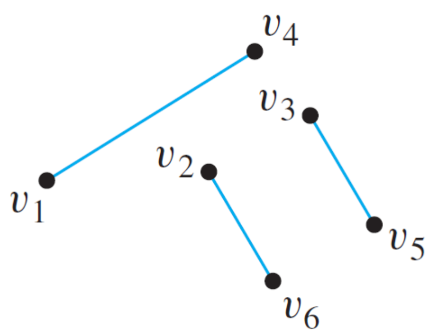
Which of the following graphs are connected?



Example 3 – *Solution*

The graph represented in (*a*) is connected, whereas those of (*b*) and (*c*) are not. To understand why (*c*) is not connected, we know that in a drawing of a graph, two edges may cross at a point that is not a vertex.

Thus the graph in (*c*) can be redrawn as follows:



Connectedness

Some useful facts relating circuits and connectedness are collected in the following lemma.

Lemma 10.2.1

Let G be a graph.

a. If G is connected, then any two distinct vertices of G can be connected by a path.

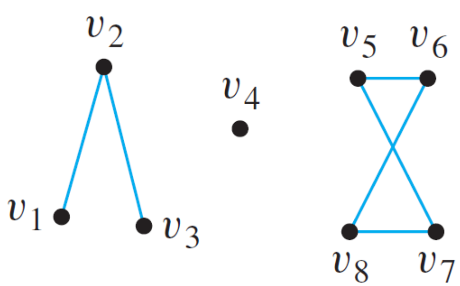
b. If vertices v and w are part of a circuit in G and one edge is removed from the circuit, then there still exists a trail from v to w in G.

c. If G is connected and G contains a circuit, then an edge of the circuit can be removed without disconnecting G.

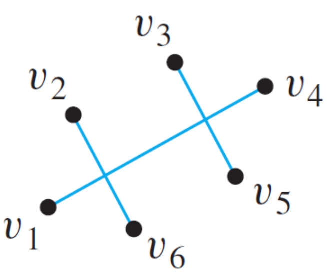
Connectedness

The graphs in (*b*) and (*c*) are both made up of three pieces, each of which is itself a connected graph.

A ***connected component*** of a graph is a connected subgraph of largest possible size.



(b)



(c)

Connectedness

- **Definition**

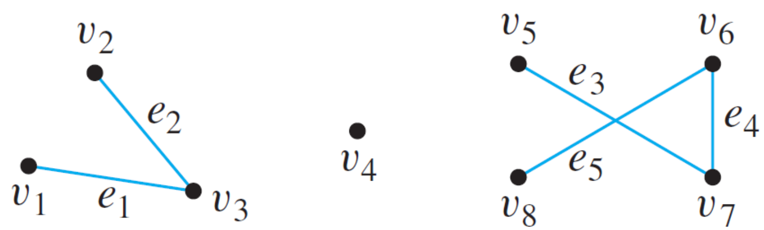
A graph H is a **connected component** of a graph G if, and only if,

1. H is subgraph of G;
2. H is connected; and
3. no connected subgraph of G has H as a subgraph and contains vertices or edges that are not in H.

The fact is that any graph is a kind of union of its connected components.

Example 4 – *Connected Components*

Find all connected components of the following graph G.



Solution:

G has three connected components: H_1, H_2, and H_3 with vertex sets V_1, V_2, and V_3 and edge sets E_1, E_2, and E_3, where

$$V_1 = \{v_1, v_2, v_3\}, \qquad E_1 = \{e_1, e_2\},$$

$$V_2 = \{v_4\}, \qquad E_2 = \emptyset,$$

$$V_3 = \{v_5, v_6, v_7, v_8\}, \qquad E_3 = \{e_3, e_4, e_5\}.$$

Euler Circuits

Euler Circuits

Now we return to consider general problems similar to the puzzle of the Königsberg bridges.

The following definition is made in honor of Euler.

• Definition

Let G be a graph. An **Euler circuit** for G is a circuit that contains every vertex and every edge of G. That is, an Euler circuit for G is a sequence of adjacent vertices and edges in G that has at least one edge, starts and ends at the same vertex, uses every vertex of G at least once, and uses every edge of G exactly once.

Euler Circuits

The analysis used earlier to solve the puzzle of the Königsberg bridges generalizes to prove the following theorem:

Theorem 10.2.2

If a graph has an Euler circuit, then every vertex of the graph has positive even degree.

Euler Circuits

We know that the contrapositive of a statement is logically equivalent to the statement.

The contrapositive of Theorem 10.2.2 is as follows:

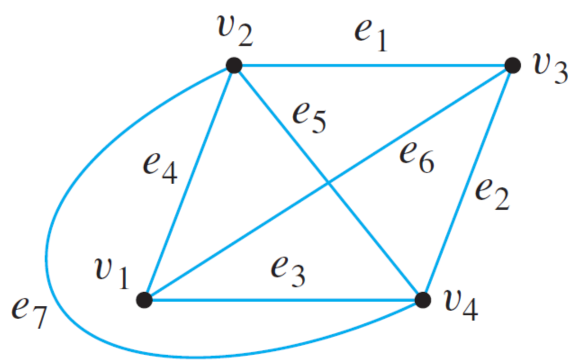
Contrapositive Version of Theorem 10.2.2

If some vertex of a graph has odd degree, then the graph does not have an Euler circuit.

This version of Theorem 10.2.2 is useful for showing that a given graph does *not* have an Euler circuit.

Example 5 – *Showing That a Graph Does Not Have an Euler Circuit*

Show that the graph below does not have an Euler circuit.



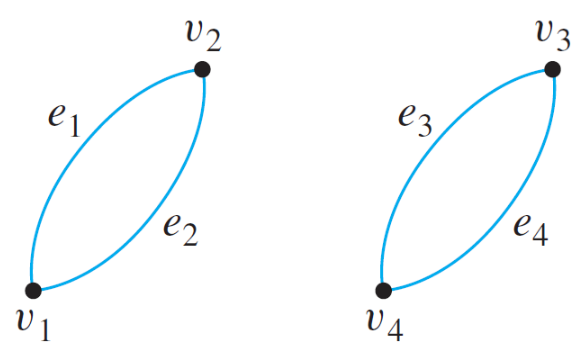
Solution:

Vertices v_1 and v_3 both have degree 3, which is odd. Hence by (the contrapositive form of) Theorem 10.2.2, this graph does not have an Euler circuit.

Euler Circuits

Now consider the converse of Theorem 10.2.2: If every vertex of a graph has even degree, then the graph has an Euler circuit. Is this true?

The answer is no. There is a graph *G* such that every vertex of *G* has even degree but *G* does not have an Euler circuit. In fact, there are many such graphs. The illustration below shows one example.



Every vertex has even degree, but the graph does not have an Euler circuit.

Euler Circuits

Note that the graph in the preceding drawing is not connected.

It turns out that although the converse of Theorem 10.2.2 is false, a modified converse is true:

If every vertex of a graph has positive even degree *and* the graph is connected, then, the graph has an Euler circuit.

Euler Circuits

Theorem 10.2.3

If a graph G is connected and the degree of every vertex of G is a positive even integer, then G has an Euler circuit.

The proof of this fact is constructive: It contains an algorithm to find an Euler circuit for any connected graph in which every vertex has even degree.

The following theorem gives a complete characterization of Euler circuits.

Theorem 10.2.4

A graph G has an Euler circuit if, and only if, G is connected and every vertex of G has positive even degree.

Euler Circuits

A corollary to Theorem 10.2.4 gives a criterion for determining when it is possible to find a walk from one vertex of a graph to another, passing through every vertex of the graph at least once and every edge of the graph exactly once.

• Definition

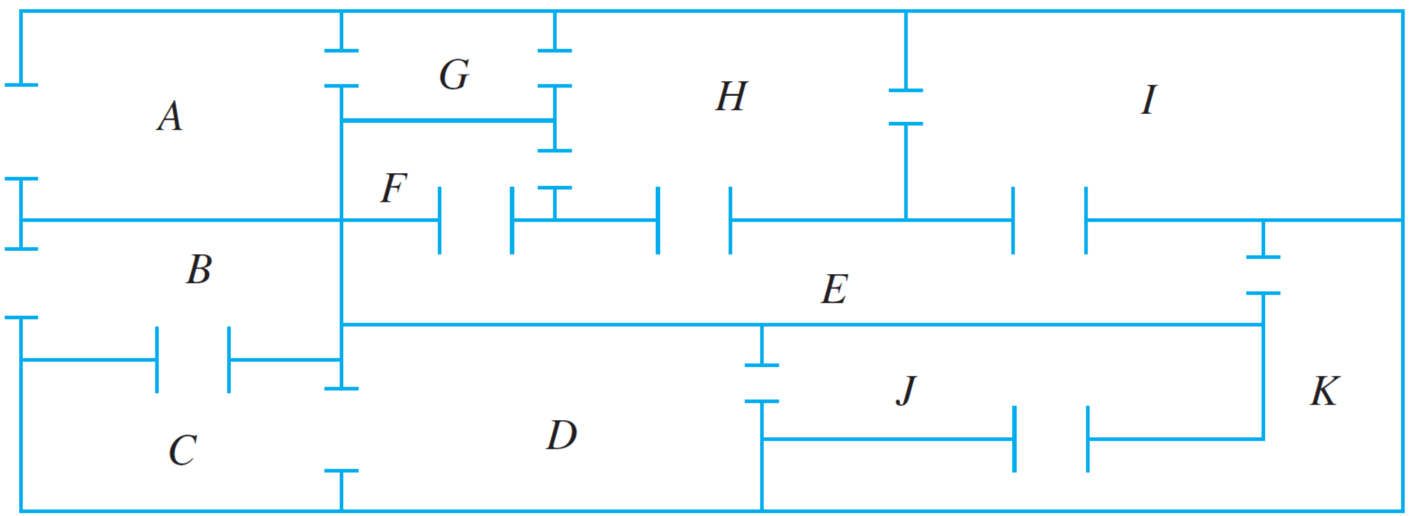
Let G be a graph, and let v and w be two distinct vertices of G. An **Euler trail from *v* to *w*** is a sequence of adjacent edges and vertices that starts at v, ends at w, passes through every vertex of G at least once, and traverses every edge of G exactly once.

Corollary 10.2.5

Let G be a graph, and let v and w be two distinct vertices of G. There is an Euler path from v to w if, and only if, G is connected, v and w have odd degree, and all other vertices of G have positive even degree.

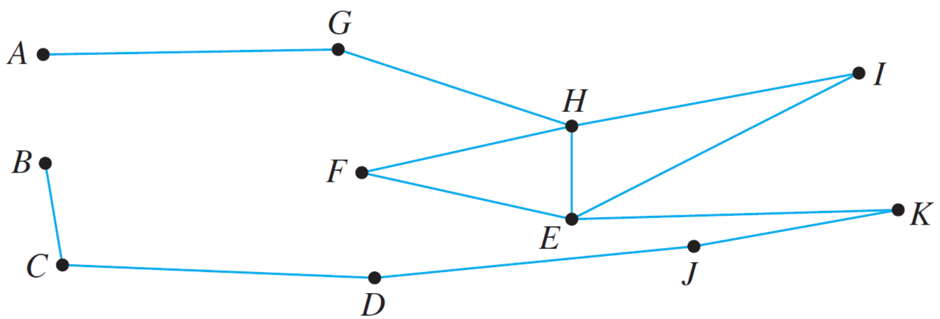
Example 7 – *Finding an Euler Trail*

The floor plan shown below is for a house that is open for public viewing. Is it possible to find a trail that starts in room *A*, ends in room *B*, and passes through every interior doorway of the house exactly once? If so, find such a trail.



Example 7 – *Solution*

Let the floor plan of the house be represented by the graph below.



Each vertex of this graph has even degree except for A and B, each of which has degree 1.

Hence by Corollary 10.2.5, there is an Euler trail from A to B. One such trail is

$$AGHFEIHEKJDCB.$$

Hamiltonian Circuits

Hamiltonian Circuits

Theorem 10.2.4 completely answers the following question: Given a graph G, is it possible to find a circuit for G in which all the *edges* of G appear exactly once?

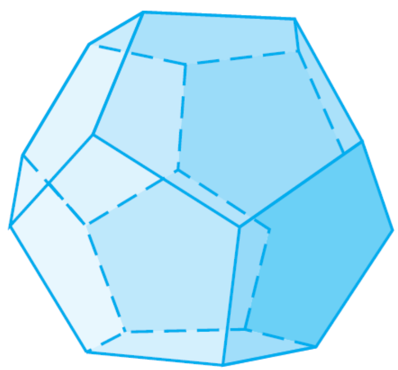
Theorem 10.2.4

A graph G has an Euler circuit if, and only if, G is connected and every vertex of G has positive even degree.

A related question is this: Given a graph G, is it possible to find a circuit for G in which all the *vertices* of G (except the first and the last) appear exactly once?

Hamiltonian Circuits

In 1859 the Irish mathematician Sir William Rowan Hamilton introduced a puzzle in the shape of a dodecahedron (a solid figure with 12 identical pentagonal faces.)



Dodecahedron

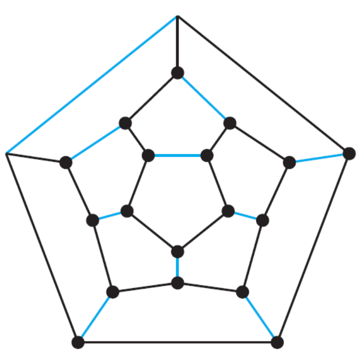
Figure 10.2.6

Hamiltonian Circuits

Each vertex was labeled with the name of a city—London, Paris, Hong Kong, New York, and so on.

The problem Hamilton posed was to start at one city and tour the world by visiting each other city exactly once and returning to the starting city.

One way to solve the puzzle is to imagine the surface of the dodecahedron stretched out and laid flat in the plane, as follows:



Hamiltonian Circuits

The circuit denoted with black lines is one solution. Note that although every city is visited, many edges are omitted from the circuit. (More difficult versions of the puzzle required that certain cities be visited in a certain order.)

The following definition is made in honor of Hamilton.

• Definition

Given a graph G, a **Hamiltonian circuit** for G is a simple circuit that includes every vertex of G. That is, a Hamiltonian circuit for G is a sequence of adjacent vertices and distinct edges in which every vertex of G appears exactly once, except for the first and the last, which are the same.

Hamiltonian Circuits

Note that although an Euler circuit for a graph *G* must include every vertex of *G*, it may visit some vertices more than once and hence may not be a Hamiltonian circuit.

On the other hand, a Hamiltonian circuit for *G* does not need to include all the edges of *G* and hence may not be an Euler circuit.

Despite the analogous-sounding definitions of Euler and Hamiltonian circuits, the mathematics of the two are very different.

Hamiltonian Circuits

Theorem 10.2.4 gives a simple criterion for determining whether a given graph has an Euler circuit.

Theorem 10.2.4

A graph G has an Euler circuit if, and only if, G is connected and every vertex of G has positive even degree.

Unfortunately, there is no analogous criterion for determining whether a given graph has a Hamiltonian circuit, nor is there even an efficient algorithm for finding such a circuit.

Hamiltonian Circuits

There is, however, a simple technique that can be used in many cases to show that a graph does *not* have a Hamiltonian circuit.

This follows from the following considerations:

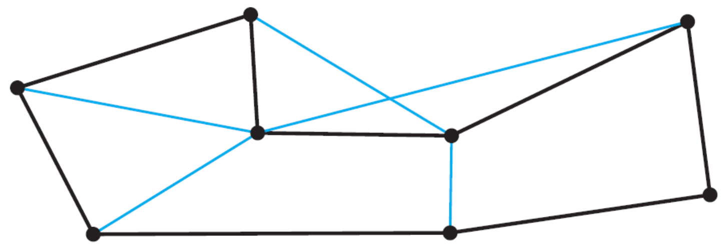
Suppose a graph *G* with at least two vertices has a Hamiltonian circuit *C* given concretely as

$$C\colon v_0 e_1 v_1 e_2 \cdots v_{n-1} e_n v_n.$$

Since *C* is a simple circuit, all the e_i are distinct and all the v_j are distinct except that $v_0 = v_n$. Let *H* be the subgraph of *G* that is formed using the vertices and edges of *C*.

Hamiltonian Circuits

An example of such an H is shown below.



H is indicated by the black lines.

Note that H has the same number of edges as it has vertices since all its n edges are distinct and so are its n vertices $v_1, v_2, \ldots, v_n$.

Also, by definition of Hamiltonian circuit, every vertex of G is a vertex of H, and H is connected since any two of its vertices lie on a circuit. In addition, every vertex of H has degree 2.

Hamiltonian Circuits

The reason for this is that there are exactly two edges incident on any vertex. These are e_i and e_{i+1} for any vertex v_i except $v_0 = v_n$, and they are e_1 and e_n for v_0 (= v_n).

These observations have established the truth of the following proposition in all cases where G has at least two vertices.

Proposition 10.2.6

If a graph G has a Hamiltonian circuit, then G has a subgraph H with the following properties:

1. H contains every vertex of G.
2. H is connected.
3. H has the same number of edges as vertices.
4. Every vertex of H has degree 2.

Hamiltonian Circuits

Note that if G contains only one vertex and G has a Hamiltonian circuit, then the circuit has the form $v\ e\ v$, where v is the vertex of G and e is an edge incident on v.

In this case, the subgraph H consisting of v and e satisfies conditions (1)–(4) of Proposition 10.2.6.

Hamiltonian Circuits

We know that the contrapositive of a statement is logically equivalent to the statement.

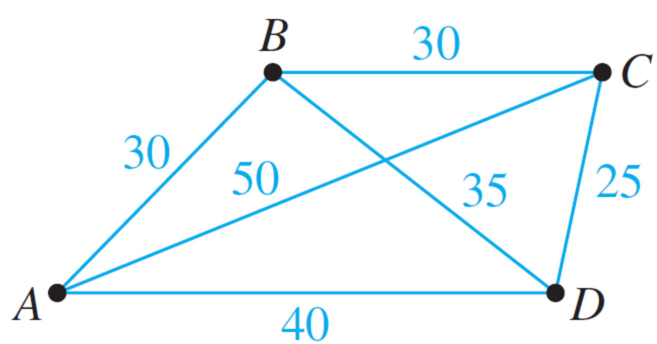
The contrapositive of Proposition 10.2.6 says that if a graph *G* does *not* have a subgraph *H* with properties (1)–(4), then *G* does *not* have a Hamiltonian circuit.

The next example illustrates a type of problem known as a **traveling salesman problem**. It is a variation of the problem of finding a Hamiltonian circuit for a graph.

Example 9 – *A Traveling Salesman Problem*

Imagine that the drawing below is a map showing four cities and the distances in kilometers between them.

Suppose that a salesman must travel to each city exactly once, starting and ending in city *A*. Which route from city to city will minimize the total distance that must be traveled?



Example 9 – *Solution*

This problem can be solved by writing all possible Hamiltonian circuits starting and ending at *A* and calculating the total distance traveled for each.

Route	Total Distance (In Kilometers)
$ABCDA$	$30 + 30 + 25 + 40 = 125$
$ABDCA$	$30 + 35 + 25 + 50 = 140$
$ACBDA$	$50 + 30 + 35 + 40 = 155$
$ACDBA$	140 [$ABDCA$ backwards]
$ADBCA$	155 [$ACBDA$ backwards]
$ADCBA$	125 [$ABCDA$ backwards]

Thus either route *ABCDA* or *ADCBA* gives a minimum total distance of 125 kilometers.

Hamiltonian Circuits

The general traveling salesman problem involves finding a Hamiltonian circuit to minimize the total distance traveled for an arbitrary graph with *n* vertices in which each edge is marked with a distance.

One way to solve the general problem is to write down all Hamiltonian circuits starting and ending at a particular vertex, compute the total distance for each, and pick one for which this total is minimal.

Hamiltonian Circuits

However, even for medium-sized values of n this method is impractical!

For a complete graph with 30 vertices, there would be $(29!)/2 \cong 4.42 \times 10^{30}$ Hamiltonian circuits starting and ending at a particular vertex to check.

Even if each circuit could be found and its total distance computed in just one nanosecond, it would require approximately **1.4×10^{14} years** to finish the computation.

Hamiltonian Circuits

At present, there is no known algorithm for solving the general traveling salesman problem that is more efficient.

However, there are efficient algorithms that find "pretty good" solutions—that is, circuits that, while not necessarily having the least possible total distances, have smaller total distances than most other Hamiltonian circuits.