Representing Graphs and Graph Isomorphism

- In this section we will show how to represent graphs in several different ways.
- Sometimes, two graphs have exactly the same form, in the sense that there is a one-to-one correspondence between their vertex sets that preserves edges. In such a case, we say that the two graphs are **isomorphic**.
- Determining whether two graphs are isomorphic is an important problem of graph theory that we will study in this section.

Representing Graphs

Representing Graphs

One way to represent a graph without multiple edges is to **list all the edges** of this graph. Another way to represent a graph with no multiple edges is to use **adjacency lists**, which specify the vertices that are adjacent to each vertex of the graph.

EXAMPLE 1

Use adjacency lists to describe the simple graph given in Figure 1.
Solution: Table 1 lists those vertices adjacent to each of the vertices of the graph

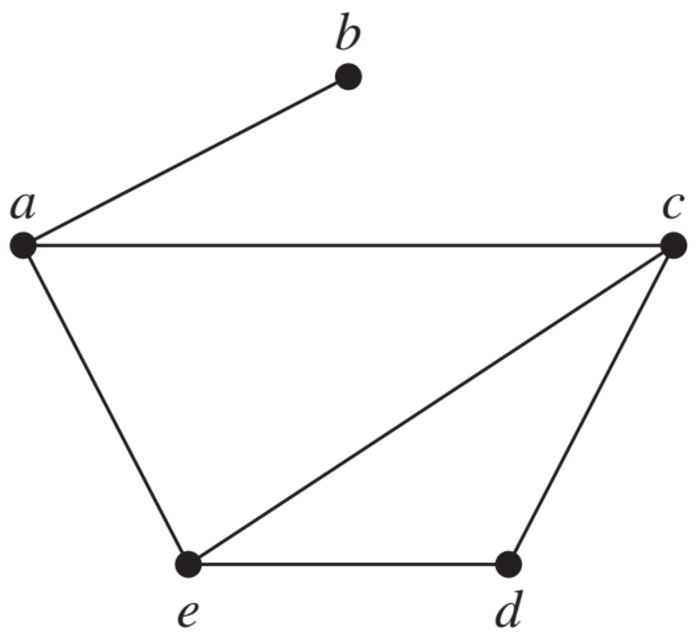

FIGURE 1 **A Simple Graph.**

TABLE 1 **An Adjacency List for a Simple Graph.**

Vertex	*Adjacent Vertices*
a	b, c, e
b	a
c	a, d, e
d	c, e
e	a, c, d

EXAMPLE 2

Represent the directed graph shown in Figure 2 by listing all the vertices that are the terminal vertices of edges starting at each vertex of the graph.

Solution:

Table 2 represents the directed graph shown in Figure 2

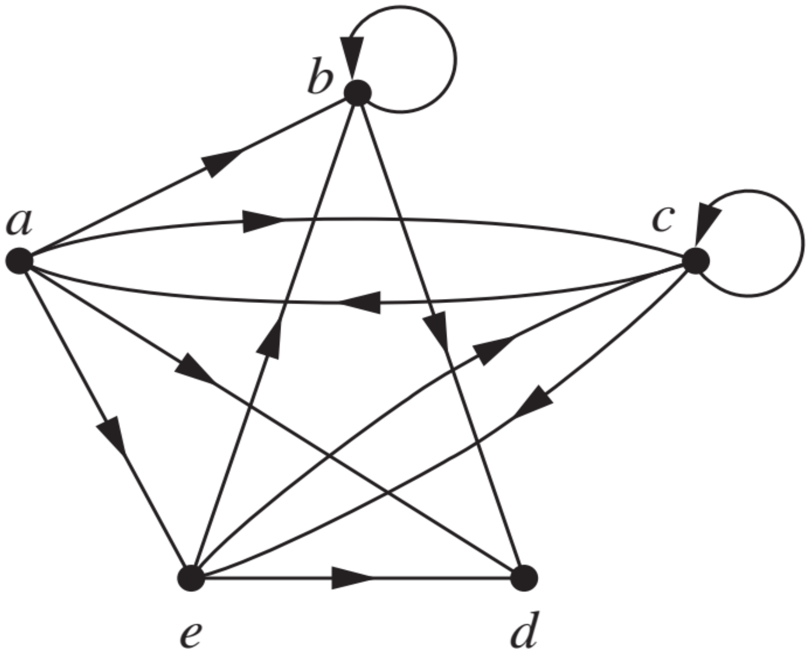

FIGURE 2 **A Directed Graph.**

TABLE 2 An Adjacency List for a Directed Graph.

Initial Vertex	*Terminal Vertices*
a	b, c, d, e
b	b, d
c	a, c, e
d	
e	b, c, d

Adjacency Matrices

- Carrying out graph algorithms using the representation of graphs by lists of edges, or by adjacency lists, can be cumbersome if there are many edges in the graph.
- To simplify computation, graphs can be represented using matrices.
- Two types of matrices commonly used to represent graphs will be presented here.
- One is based on the adjacency of vertices, and the other is based on incidence of vertices and edges.

Suppose that $G = (V, E)$ is a simple graph where $|V| = n$. Suppose that the vertices of G are listed arbitrarily as $v_1, v_2, \ldots, v_n$. The **adjacency matrix A** (or $\mathbf{A}_G$) of G, with respect to this listing of the vertices, is the n x n zero–one matrix with 1 as its (i, j)th entry when v_i and v_j are adjacent, and 0 as its (i, j)th entry when they are not adjacent. In other words, if its adjacency matrix is $\mathbf{A} = [a_{ij}]$, then

$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \text{ is an edge of } G, \\ 0 & \text{otherwise.} \end{cases}$$

EXAMPLE 3

Use an adjacency matrix to represent the graph shown in Figure 3.

Solution: We order the vertices as a, b, c, d. The matrix representing this graph is

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

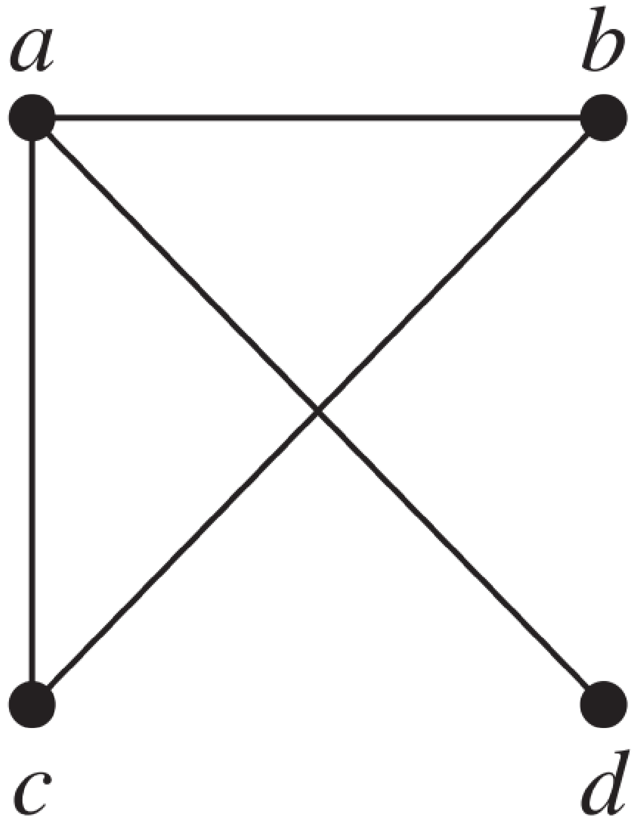

FIGURE 3
Simple Graph.

EXAMPLE 4 Draw a graph with the adjacency matrix

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

with respect to the ordering of vertices a, b, c, d.

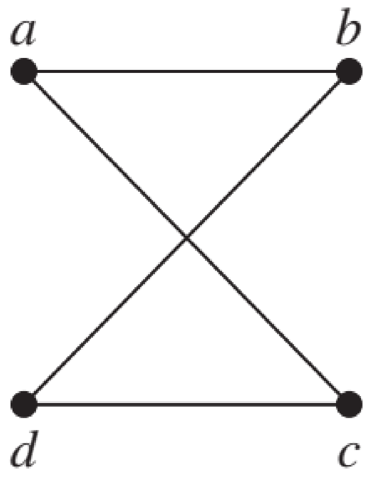

FIGURE 4
A Graph with the Given Adjacency Matrix.

Solution: A graph with this adjacency matrix is shown in Figure 4.

Note that an adjacency matrix of a graph is based on the ordering chosen for the vertices. Hence, there may be as many as $n!$ different adjacency matrices for a graph with n vertices, because there are $n!$ different orderings of n vertices.

The adjacency matrix of a simple graph is symmetric, that is, $a_{ij} = a_{ji}$, because both of these entries are 1 when v_i and v_j are adjacent, and both are 0 otherwise. Furthermore, because a simple graph has no loops, each entry a_{ii}, $i = 1, 2, 3, \ldots, n$, is 0.

Adjacency matrices can also be used to represent undirected graphs with loops and with multiple edges. A loop at the vertex v_i is represented by a 1 at the (i, i)th position of the adjacency matrix. When multiple edges connecting the same pair of vertices v_i and v_j, or multiple loops at the same vertex, are present, the adjacency matrix is no longer a zero – one matrix, because the (i, j)th entry of this matrix equals the number of edges that are associated to $\{v_i, v_j\}$. All undirected graphs, including multigraphs and pseudographs, have symmetric adjacency matrices.

EXAMPLE 5

Use an adjacency matrix to represent the pseudograph shown in Figure 5.

Solution: The adjacency matrix using the ordering of vertices a, b, c, d is

$$\begin{bmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix}.$$

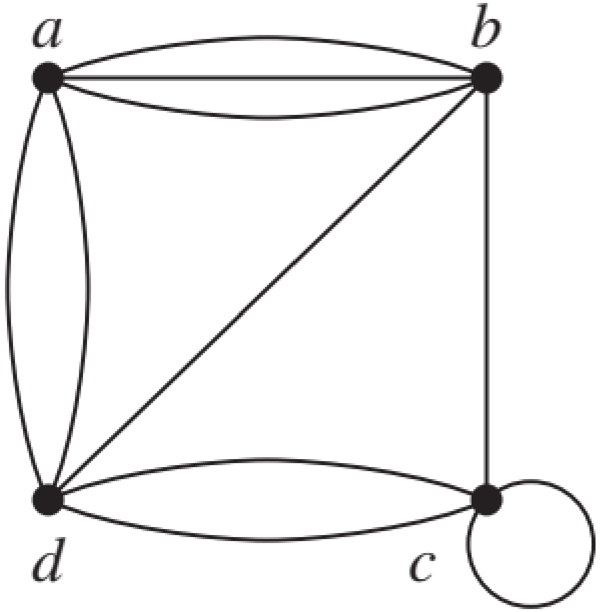

FIGURE 5
A Pseudograph.

The matrix for a directed graph

- We used zero–one matrices in Chapter 9 to represent directed graphs. The matrix for a directed graph $G = (V, E)$ has a 1 in its (i, j)th position if there is an edge from v_i to v_j, where $v_1, v_2, \ldots, v_n$ is an arbitrary listing of the vertices of the directed graph.

- In other words, if $\boldsymbol{A} = [a_{ij}]$ is the adjacency matrix for the directed graph with respect to this listing of the vertices, then

$$a_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \text{ is an edge of } G, \\ 0 & \text{otherwise.} \end{cases}$$

The matrix for a directed graph (Continue)

- The adjacency matrix for a directed graph does not have to be symmetric, because there may not be an edge from v_j to v_i when there is an edge from v_i to v_j.
- Adjacency matrices can also be used to represent directed multigraphs. Again, such matrices are not zero–one matrices when there are multiple edges in the same direction connecting two vertices. In the adjacency matrix for a directed multigraph, a_{ij} equals the number of edges that are associated to (v_i, v_j).

TRADE-OFFS BETWEEN ADJACENCY LISTS AND ADJACENCY MATRICES

- When a simple graph contains relatively few edges, that is, when it is **sparse**, it is usually preferable to use adjacency lists rather than an adjacency matrix to represent the graph. For example, if each vertex has degree not exceeding c, where c is a constant much smaller than n, then each adjacency list contains c or fewer vertices. Hence, there are no more than cn items in all these adjacency lists. On the other hand, the adjacency matrix for the graph has n^2 entries. Note, however, that the adjacency matrix of a sparse graph is a **sparse matrix**, that is, a matrix with few nonzero entries, and there are special techniques for representing, and computing with, sparse matrices.

TRADE-OFFS BETWEEN ADJACENCY LISTS AND ADJACENCY MATRICES (Continue)

- Now suppose that a simple graph is **dense**, that is, suppose that it contains many edges, such as a graph that contains more than half of all possible edges. In this case, using an adjacency matrix to represent the graph is usually preferable over using adjacency lists. To see why, we compare the complexity of determining whether the possible edge $\{v_i, v_j\}$ is present. Using an adjacency matrix, we can determine whether this edge is present by examining the (i, j)th entry in the matrix. This entry is 1 if the graph contains this edge and is 0 otherwise.
- Consequently, we need make only one comparison, namely, comparing this entry with 0, to determine whether this edge is present. On the other hand, when we use adjacency lists to represent the graph, we need to search the list of vertices adjacent to either v_i or v_j to determine whether this edge is present. This can require $\Theta(|V|)$ comparisons when many edges are present.

Incidence Matrices

Another common way to represent graphs is to use **incidence matrices**. Let $G = (V, E)$ be an undirected graph. Suppose that $v_1, v_2, \ldots, v_n$ are the vertices and $e_1, e_2, \ldots, e_m$ are the edges of G. Then the incidence matrix with respect to this ordering of V and E is the $n \times m$ matrix $\mathbf{M} = [m_{ij}]$, where

$$m_{ij} = \begin{cases} 1 & \text{when edge } e_j \text{ is incident with } v_i, \\ 0 & \text{otherwise.} \end{cases}$$

EXAMPLE 6

Represent the graph shown in Figure 6 with an incidence matrix.

Solution: The incidence matrix is

$$
\begin{array}{c c}
 & \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} &
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0
\end{bmatrix}
\end{array}.
$$

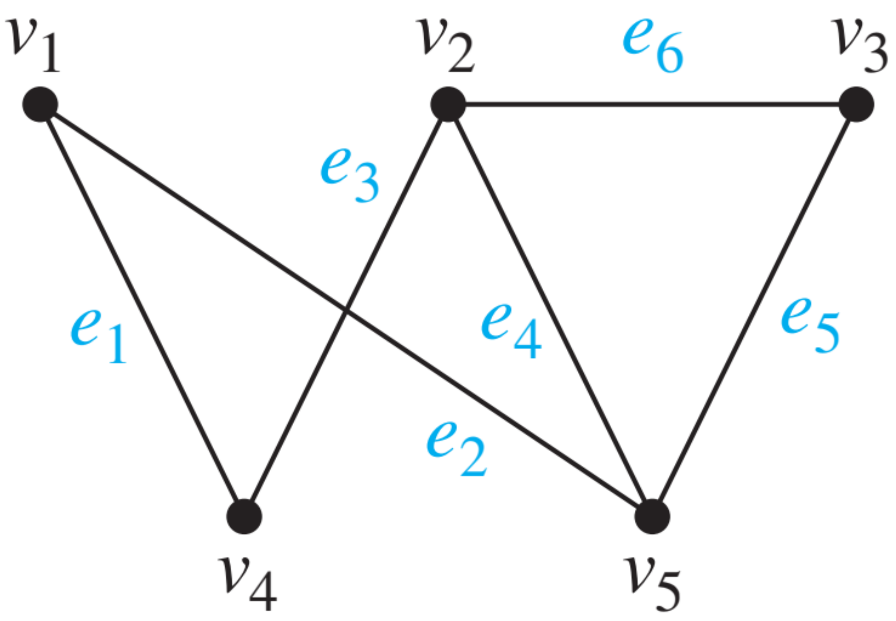

FIGURE 6 An Undirected Graph.

Incidence matrices can also be used to represent multiple edges and loops. Multiple edges are represented in the incidence matrix using columns with identical entries, because these edges are incident with the same pair of vertices. Loops are represented using a column with exactly one entry equal to 1, corresponding to the vertex that is incident with this loop.

EXAMPLE 7 Represent the pseudograph shown in Figure 7 using an incidence matrix.

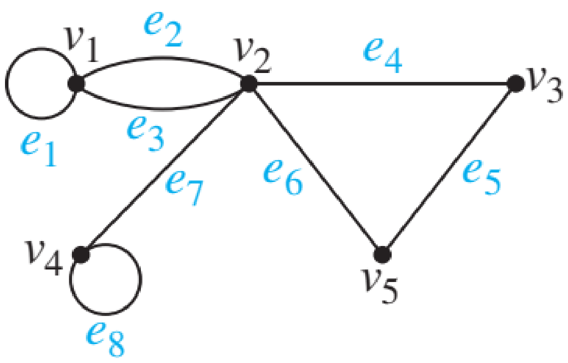

FIGURE 7
A Pseudograph.

Solution: The incidence matrix for this graph is

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \end{array}.$$

Isomorphism of Graphs

- We often need to know whether it is possible to draw two graphs in the same way. That is, do the graphs have the same structure when we ignore the identities of their vertices?
- For instance, in chemistry, graphs are used to model chemical compounds (in a way we will describe later).
- Different compounds can have the same molecular formula but can differ in structure. Such compounds can be represented by graphs that cannot be drawn in the same way.
- The graphs representing previously known compounds can be used to determine whether a supposedly new compound has been studied before.

Isomorphism of Graphs (Continue)

DEFINITION 1

The simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are *isomorphic* if there exists a one-to-one and onto function f from V_1 to V_2 with the property that a and b are adjacent in G_1 if and only if $f(a)$ and $f(b)$ are adjacent in G_2, for all a and b in V_1. Such a function f is called an *isomorphism*.*

Note:

- Two simple graphs that are not isomorphic are called *nonisomorphic*.
- In other words, when two simple graphs are isomorphic, there is a one-to-one correspondence between vertices of the two graphs that preserves the adjacency relationship. Isomorphism of simple graphs is an equivalence relation.

EXAMPLE 8

Show that the graphs $G = (V, E)$ and $H = (W, F)$, displayed in Figure 8, are isomorphic.

Solution: The function f with $f(u_1) = v_1$, $f(u_2) = v_4$, $f(u_3) = v_3$, and $f(u_4) = v_2$ is a one-to-one correspondence between V and W. To see that this correspondence preserves adjacency, note that adjacent vertices in G are u_1 and u_2, u_1 and u_3, u_2 and u_4, and u_3 and u_4, and each of the pairs $f(u_1) = v_1$ and $f(u_2) = v_4$, $f(u_1) = v_1$ and $f(u_3) = v_3$, $f(u_2) = v_4$ and $f(u_4) = v_2$, and $f(u_3) = v_3$ and $f(u_4) = v_2$ consists of two adjacent vertices in H. ◀

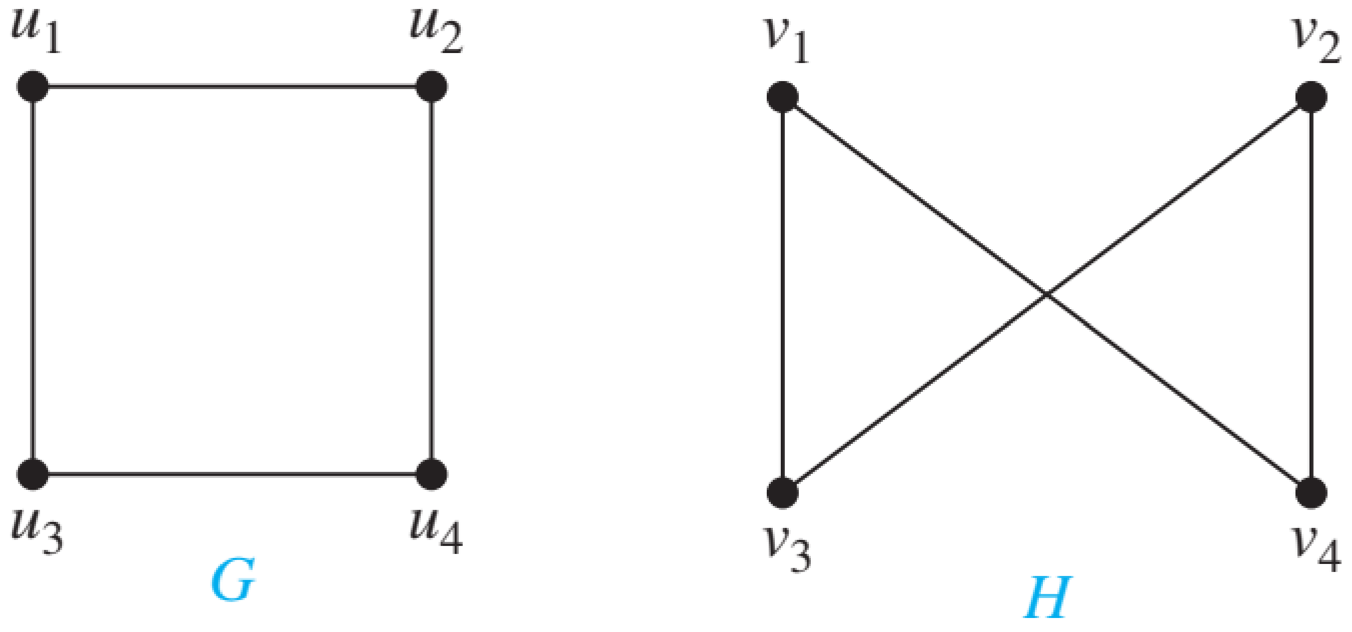

FIGURE 8 **The Graphs G and H.**

Determining whether Two Simple Graphs are Isomorphic

- It is often difficult to determine whether two simple graphs are isomorphic. There are $n!$ possible one-to-one correspondences between the vertex sets of two simple graphs with n vertices. Testing each such correspondence to see whether it preserves adjacency and nonadjacency is impractical if n is at all large.
- In particular, we can show that two graphs are not isomorphic if we can find a property only one of the two graphs has, but that is preserved by isomorphism.
- A property preserved by isomorphism of graphs is called a **graph invariant**. For instance, isomorphic simple graphs must have the same number of vertices, because there is a one-to-one correspondence between the sets of vertices of the graphs.

Determining whether Two Simple Graphs are Isomorphic (Continue)

- Isomorphic simple graphs also must have the same number of edges, because the one-to-one correspondence between vertices establishes a one-to-one correspondence between edges.
- In addition, the degrees of the vertices in isomorphic simple graphs must be the same. That is, a vertex v of degree d in G must correspond to a vertex $f(v)$ of degree d in H, because a vertex w in G is adjacent to v if and only if $f(v)$ and $f(w)$ are adjacent in H.

Determining whether Two Simple Graphs are Isomorphic (Continue)

EXAMPLE 9

Show that the graphs displayed in Figure 9 are not isomorphic.

Solution: Both G and H have five vertices and six edges. However, H has a vertex of degree one, namely, e, whereas G has no vertices of degree one. It follows that G and H are not isomorphic

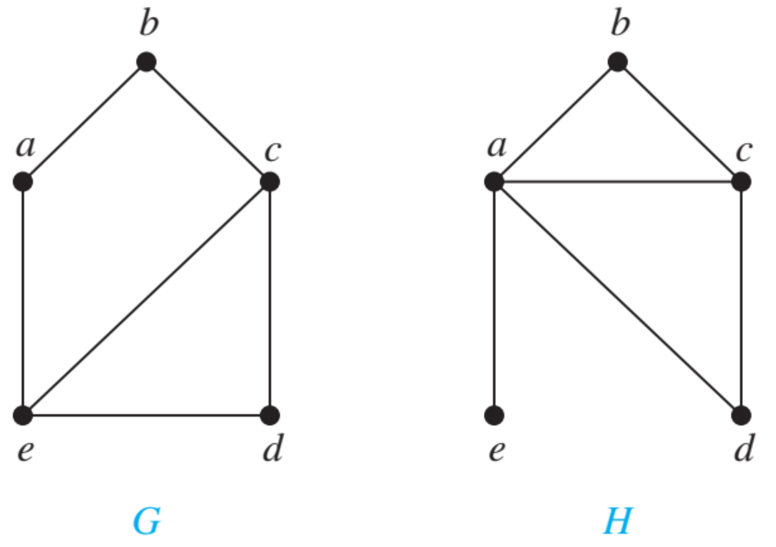

FIGURE 9 The Graphs G and H.

Determine whether the graphs shown in Figure 10 are isomorphic.

Solution: The graphs G and H both have eight vertices and 10 edges. They also both have four vertices of degree two and four of degree three. Because these invariants all agree, it is still conceivable that these graphs are isomorphic.

However, G and H are not isomorphic. To see this, note that because $\deg(a) = 2$ in G, a must correspond to either t, u, x, or y in H, because these are the vertices of degree two in H. However, each of these four vertices in H is adjacent to another vertex of degree two in H, which is not true for a in G.

Another way to see that G and H are not isomorphic is to note that the subgraphs of G and H made up of vertices of degree three and the edges connecting them must be isomorphic if these two graphs are isomorphic (the reader should verify this). However, these subgraphs, shown in Figure 11, are not isomorphic. ◀

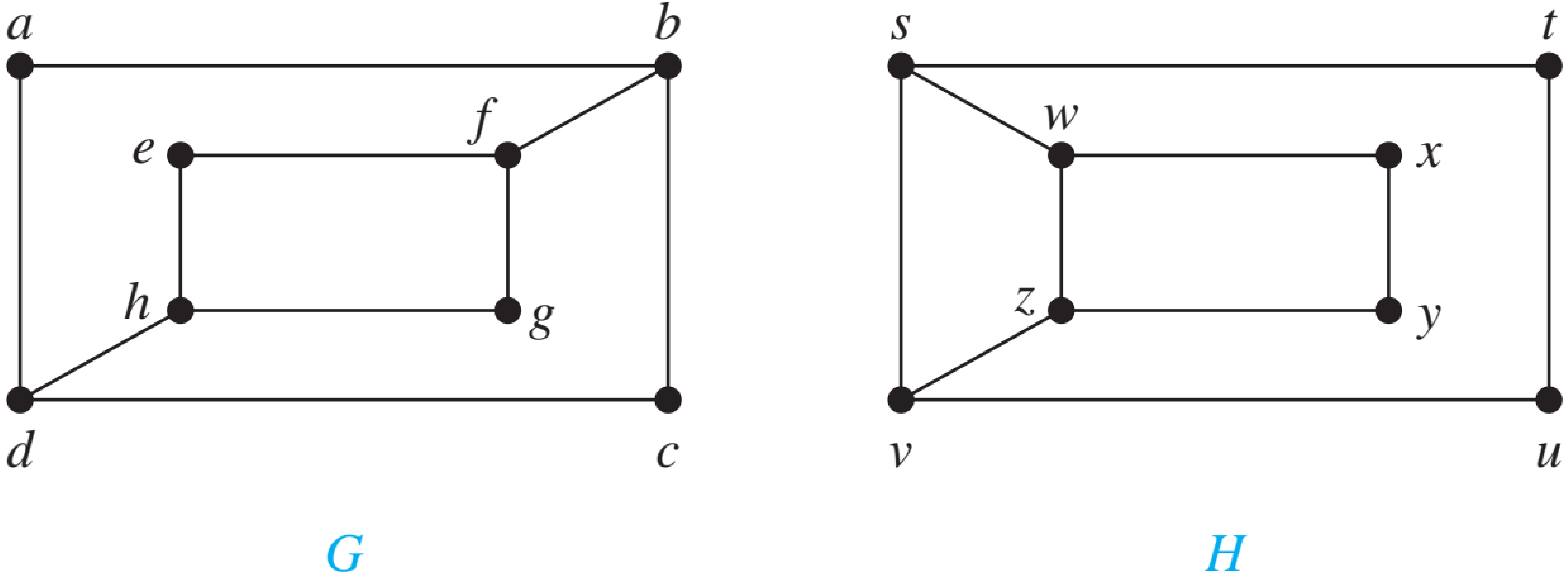

FIGURE 10 **The Graphs *G* and *H*.**

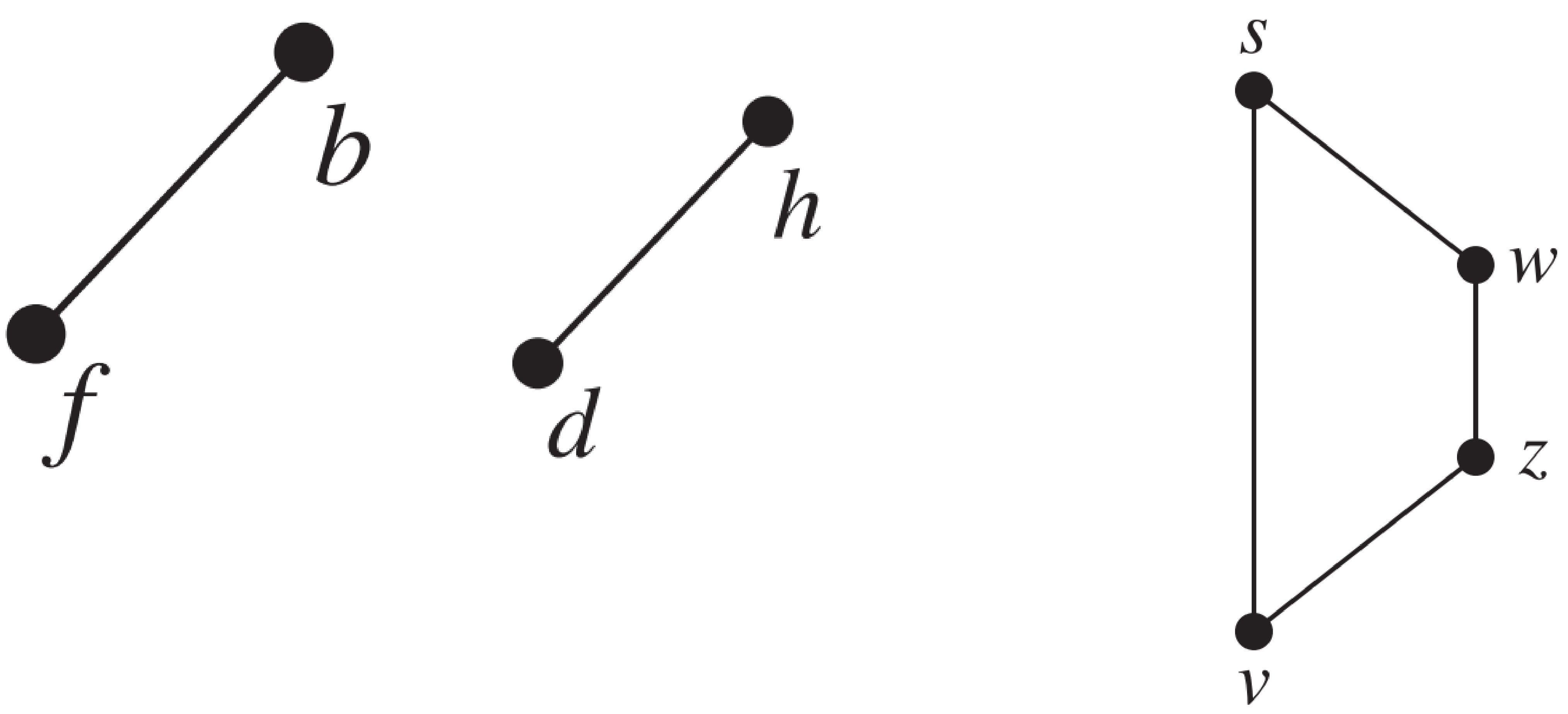

FIGURE 11 **The Subgraphs of *G* and *H* Made Up of Vertices of Degree Three and the Edges Connecting Them.**

EXAMPLE 11

Determine whether the graphs G and H displayed in Figure 12 are isomorphic.

Solution: Both G and H have six vertices and seven edges. Both have four vertices of degree two and two vertices of degree three. It is also easy to see that the subgraphs of G and H consisting of all vertices of degree two and the edges connecting them are isomorphic (as the reader should verify). Because G and H agree with respect to these invariants, it is reasonable to try to find an isomorphism f.

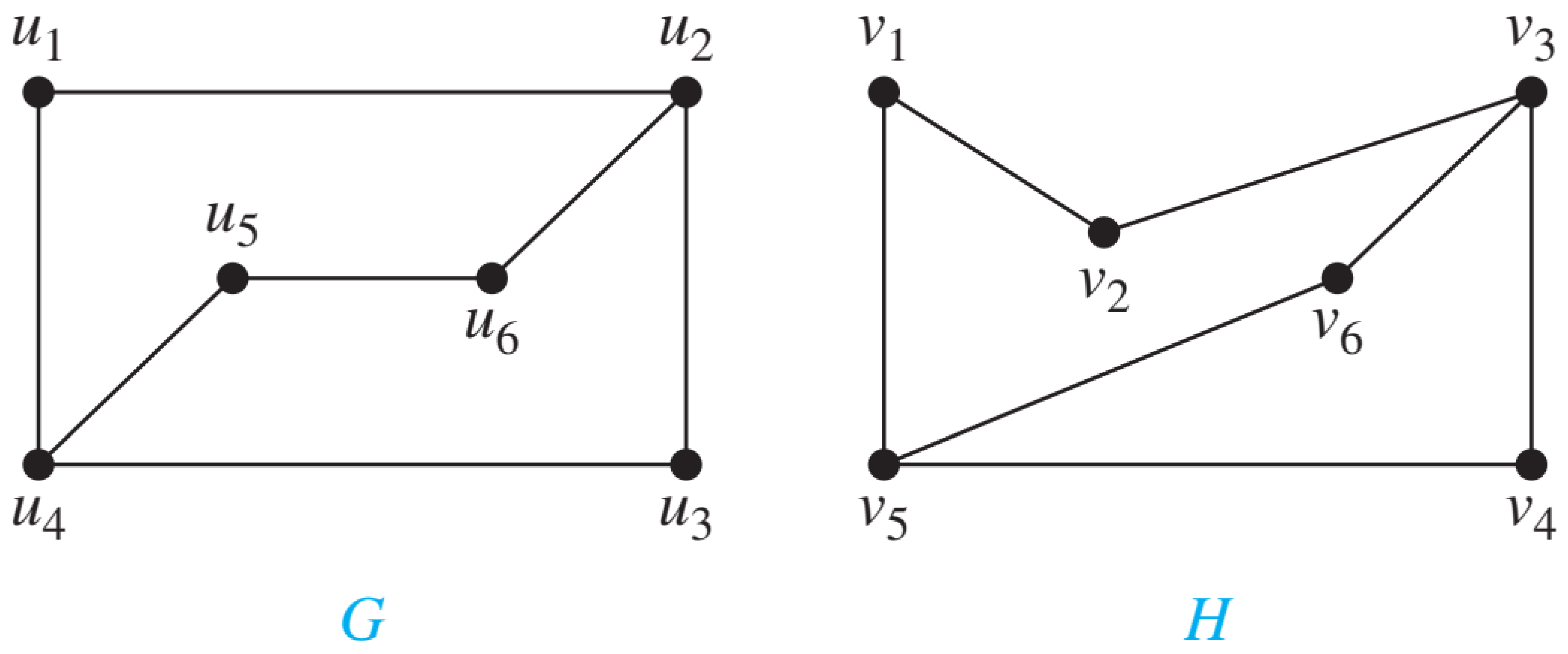

FIGURE 12 **Graphs G and H.**

We now will define a function f and then determine whether it is an isomorphism. Because $\deg(u_1) = 2$ and because u_1 is not adjacent to any other vertex of degree two, the image of u_1 must be either v_4 or v_6, the only vertices of degree two in H not adjacent to a vertex of degree two. We arbitrarily set $f(u_1) = v_6$. [If we found that this choice did not lead to isomorphism, we would then try $f(u_1) = v_4$.] Because u_2 is adjacent to u_1, the possible images of u_2 are v_3 and v_5. We arbitrarily set $f(u_2) = v_3$. Continuing in this way, using adjacency of vertices and degrees as a guide, we set $f(u_3) = v_4$, $f(u_4) = v_5$, $f(u_5) = v_1$, and $f(u_6) = v_2$. We now have a one-to-one correspondence between the vertex set of G and the vertex set of H, namely, $f(u_1) = v_6$, $f(u_2) = v_3$, $f(u_3) = v_4$, $f(u_4) = v_5$, $f(u_5) = v_1$, $f(u_6) = v_2$. To see whether f preserves edges, we examine the adjacency matrix of G,

$$\mathbf{A}_G = \begin{array}{c c} & \begin{array}{cccccc} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \end{array} \\ \begin{array}{c} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{array} & \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array},$$

and the adjacency matrix of H with the rows and columns labeled by the images of the corresponding vertices in G,

$$\mathbf{A}_H = \begin{array}{c} \\ v_6 \\ v_3 \\ v_4 \\ v_5 \\ v_1 \\ v_2 \end{array} \begin{array}{c} \begin{array}{cccccc} v_6 & v_3 & v_4 & v_5 & v_1 & v_2 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}.$$

Because $\mathbf{A}_G = \mathbf{A}_H$, it follows that f preserves edges. We conclude that f is an isomorphism, so G and H are isomorphic. Note that if f turned out not to be an isomorphism, we would *not* have established that G and H are not isomorphic, because another correspondence of the vertices in G and H may be an isomorphism. ◀