61. If the simple graph G has v vertices and e edges, how many edges does $\overline{G}$ have?

62. If the degree sequence of the simple graph G is 4, 3, 3, 2, 2, what is the degree sequence of $\overline{G}$?

63. If the degree sequence of the simple graph G is $d_1, d_2, \ldots, d_n$, what is the degree sequence of $\overline{G}$?

***64.** Show that if G is a bipartite simple graph with v vertices and e edges, then $e \leq v^2/4$.

65. Show that if G is a simple graph with n vertices, then the union of G and $\overline{G}$ is K_n.

***66.** Describe an algorithm to decide whether a graph is bipartite based on the fact that a graph is bipartite if and only if it is possible to color its vertices two different colors so that no two vertices of the same color are adjacent.

The **converse** of a directed graph $G = (V, E)$, denoted by G^{conv}, is the directed graph (V, F), where the set F of edges of G^{conv} is obtained by reversing the direction of each edge in E.

67. Draw the converse of each of the graphs in Exercises 7–9 in Section 10.1.

68. Show that $(G^{conv})^{conv} = G$ whenever G is a directed graph.

69. Show that the graph G is its own converse if and only if the relation associated with G (see Section 9.3) is symmetric.

70. Show that if a bipartite graph $G = (V, E)$ is n-regular for some positive integer n (see the preamble to Exercise 53) and (V_1, V_2) is a bipartition of V, then $|V_1| = |V_2|$. That is, show that the two sets in a bipartition of the vertex set of an n-regular graph must contain the same number of vertices.

71. Draw the mesh network for interconnecting nine parallel processors.

72. In a variant of a mesh network for interconnecting $n = m^2$ processors, processor $P(i, j)$ is connected to the four processors $P((i \pm 1) \textbf{ mod } m, j)$ and $P(i, (j \pm 1) \textbf{ mod } m)$, so that connections wrap around the edges of the mesh. Draw this variant of the mesh network for 16 processors.

73. Show that every pair of processors in a mesh network of $n = m^2$ processors can communicate using $O(\sqrt{n}) = O(m)$ hops between directly connected processors.

10.3 Representing Graphs and Graph Isomorphism

12/13 14:24
| read
15:17
| notes

Introduction

There are many useful ways to represent graphs. As we will see throughout this chapter, in working with a graph it is helpful to be able to choose its most convenient representation. 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

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. ◀

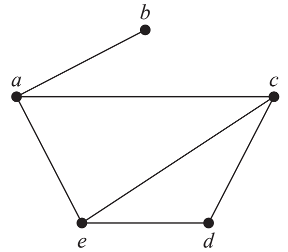


FIGURE 1 **A Simple Graph.**

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

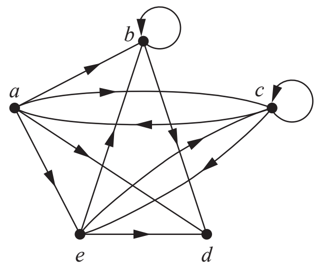


FIGURE 2 **A Directed Graph.**

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

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

Solution: Table 2 represents the directed graph shown in Figure 2. ◀

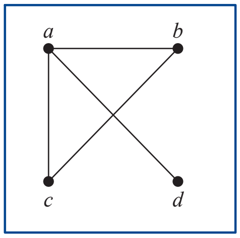


FIGURE 3
Simple Graph.

Adjacency Matrices

Links

Carrying out graph algorithms using the representation of graphs by lists of edges, or by adjacency lists, can be cumbersome (slow or complicated and therefore inefficient) 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 $n \times n$ 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

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

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

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

◀

EXAMPLE 4 Draw a graph with the adjacency matrix

$$\begin{array}{cc} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

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

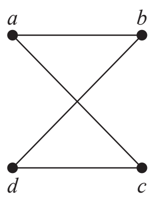


FIGURE 4
A Graph with the Given Adjacency Matrix.

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.

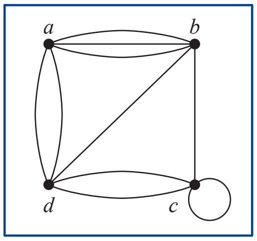


FIGURE 5
A Pseudograph.

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

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

◀

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 $\mathbf{A} = [a_{ij}]$ 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 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 稀疏，稀少的 **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.

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.

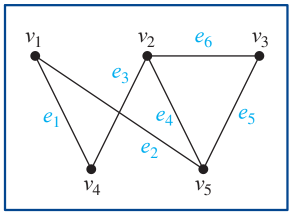


FIGURE 6 An Undirected Graph.

Solution: The incidence matrix is

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

same column connect

◀

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.

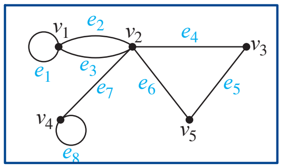


FIGURE 7 A Pseudograph.

Solution: The incidence matrix for this graph is

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

loop   connect to itself   loop

◀

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.

There is a useful terminology for graphs with the same structure.

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*.* 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. (We leave the verification of this as Exercise 45.)

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

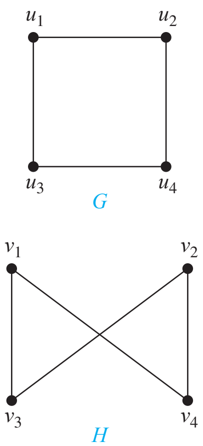


FIGURE 8 The Graphs G and H.

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. ◀

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.

Sometimes it is not hard to show that two graphs are not isomorphic. 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.

Links

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.

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

Extra Examples

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. ◀

The number of vertices, the number of edges, and the number of vertices of each degree are all invariants under isomorphism. If any of these quantities differ in two simple graphs, these graphs cannot be isomorphic. However, when these invariants are the same, it does not necessarily mean that the two graphs are isomorphic. There are no useful sets of invariants currently known that can be used to determine whether simple graphs are isomorphic.

*The word *isomorphism* comes from the Greek roots *isos* for "equal" and *morphe* for "form."

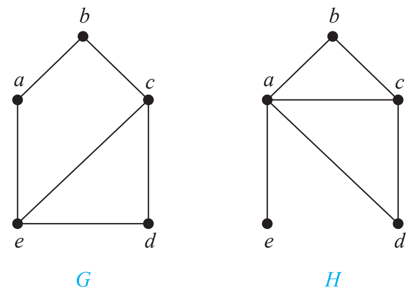


FIGURE 9 **The Graphs *G* and *H*.**

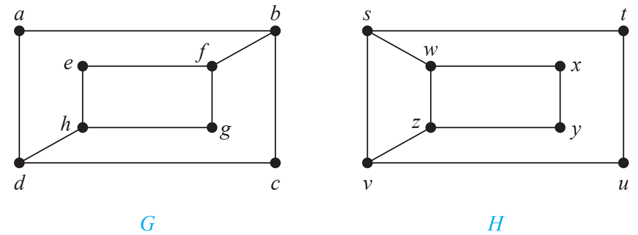


FIGURE 10 **The Graphs *G* and *H*.**

EXAMPLE 10 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. ◀

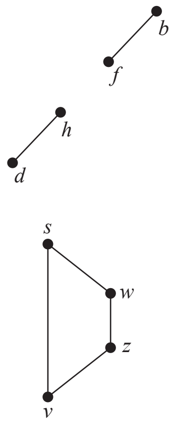
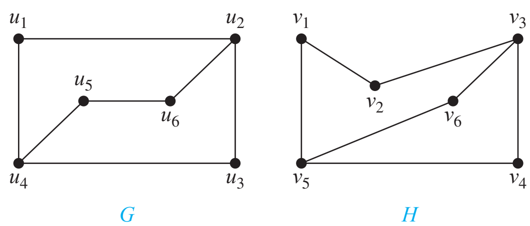


FIGURE 11 **The Subgraphs of *G* and *H* Made Up of Vertices of Degree Three and the Edges Connecting Them.**

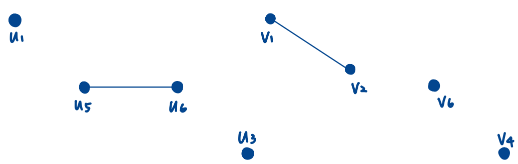
To show that a function f from the vertex set of a graph G to the vertex set of a graph H is an isomorphism, we need to show that f preserves the presence and absence of edges. One helpful way to do this is to use adjacency matrices. In particular, to show that f is an isomorphism, we can show that the adjacency matrix of G is the same as the adjacency matrix of H, when rows and columns are labeled to correspond to the images under f of the vertices in G that are the labels of these rows and columns in the adjacency matrix of G. We illustrate how this is done in Example 11.

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.



yes !?



→ subgraphs of degree 2

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,

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

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. ◀

Links

ALGORITHMS FOR GRAPH ISOMORPHISM The best algorithms known for determining whether two graphs are isomorphic have exponential worst-case time complexity (in the number of vertices of the graphs). However, linear average-case time complexity algorithms are known that solve this problem, and there is some hope, but also 懷疑論/的態度 skepticism (doubt as to the truth of something), that an algorithm with polynomial worst-case time complexity for determining whether two graphs are isomorphic can be found. The best practical general purpose software for isomorphism testing, called NAUTY, can be used to determine whether two graphs with as many as 100 vertices are isomorphic in less than a second on a modern PC. NAUTY software can be downloaded over the Internet and experimented with. Practical algorithms for determining whether two graphs are isomorphic exist for graphs that are restricted in various ways, such as when the maximum degree of vertices is small. The problem of determining whether any two graphs are isomorphic is of special interest because it is one of only a few NP problems (see Exercise 72) not known to be either 易處理的 tractable ((of a situation or problem) easy to deal with) or NP-complete (see Section 3.3).

APPLICATIONS OF GRAPH ISOMORPHISMS Graph isomorphisms, and functions that are almost graph isomorphisms, arise in applications of graph theory to chemistry and to the design of electronic circuits, and other areas including bioinformatics and computer vision.

Chemists use multigraphs, known as molecular graphs, to model chemical compounds. In these graphs, vertices represent atoms and edges represent chemical bonds between these atoms. Two structural isomers, molecules with identical molecular formulas but with atoms bonded differently, have nonisomorphic molecular graphs. When a potentially new chemical compound is synthesized, a database of molecular graphs is checked to see whether the molecular graph of the compound is the same as one already known.

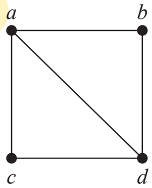
Electronic circuits are modeled using graphs in which vertices represent components and edges represent connections between them. Modern integrated circuits, known as chips, are miniaturized electronic circuits, often with millions of transistors and connections between them. Because of the complexity of modern chips, automation tools are used to design them. Graph isomorphism is the basis for the verification that a particular layout of a circuit produced by an automated tool corresponds to the original schematic of the design. Graph isomorphism can also be used to determine whether a chip from one vendor includes intellectual property from a different vendor. This can be done by looking for large isomorphic subgraphs in the graphs modeling these chips.

12/13 17:45 | do 18:40 | correct 18:58

Exercises

In Exercises 1–4 use an adjacency list to represent the given graph.

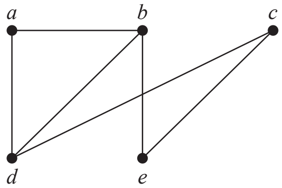
1.



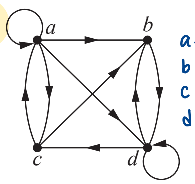
a: b, c, d
b: a, d
c: a, d
d: a, b, c

5.
$$\begin{array}{c|c} & a\ b\ c\ d \\ \begin{array}{c} a\\ b\\ c\\ d\end{array} & \begin{bmatrix} 0&1&1&1\\ 1&0&0&1\\ 1&0&0&1\\ 1&1&1&0 \end{bmatrix}\end{array}$$

2.



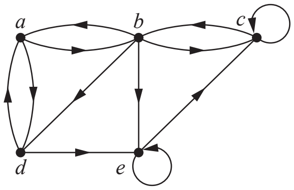
3.



a: a, b, c, d
b: d
c: a, b
d: b, c, d

7.
$$\begin{array}{c|c} & a\ b\ c\ d \\ \begin{array}{c} a\\ b\\ c\\ d\end{array} & \begin{bmatrix} 1&1&1&1\\ 0&0&0&1\\ 1&1&0&0\\ 0&1&1&1 \end{bmatrix}\end{array}$$

4.



5. Represent the graph in Exercise 1 with an adjacency matrix.

6. Represent the graph in Exercise 2 with an adjacency matrix.

7. Represent the graph in Exercise 3 with an adjacency matrix.

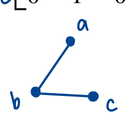
8. Represent the graph in Exercise 4 with an adjacency matrix.

9. Represent each of these graphs with an adjacency matrix.

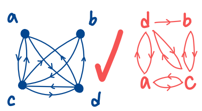
a) K_4 b) $K_{1,4}$ c) $K_{2,3}$
d) C_4 e) W_4 f) Q_3

In Exercises 10–12 draw a graph with the given adjacency matrix.

10.
$$\begin{array}{c|c} & a\ \ b\ \ c \\ \begin{array}{c} a\\ b\\ c\end{array} & \begin{bmatrix} 0&1&0\\ 1&0&1\\ 0&1&0 \end{bmatrix}\end{array}$$



11.
$$\begin{array}{c|c} & a\ \ b\ \ c\ \ d \\ \begin{array}{c} a\\ b\\ c\\ d\end{array} & \begin{bmatrix} 0&0&1&1\\ 0&0&1&0\\ 1&1&0&1\\ 1&1&1&0 \end{bmatrix}\end{array}$$

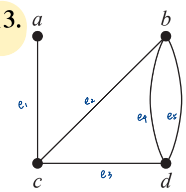


12.
$$\begin{bmatrix} 1&1&1&0\\ 0&0&1&0\\ 1&0&1&0\\ 1&1&1&0 \end{bmatrix}$$

In Exercises 13–15 represent the given graph using an adjacency matrix.

27. incidence matrix

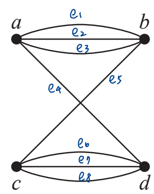
13.



$$\begin{array}{c|c} & a\ b\ c\ d \\ \begin{array}{c} a\\ b\\ c\\ d\end{array} & \begin{bmatrix} 0&0&1&0\\ 0&0&1&2\\ 1&1&0&1\\ 0&2&1&0 \end{bmatrix}\end{array}$$

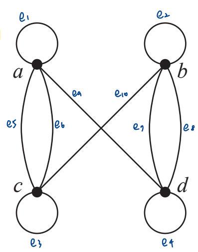
13.
$$\begin{array}{c|c} & e_1\ e_2\ e_3\ e_4\ e_5 \\ \begin{array}{c} a\\ b\\ c\\ d\end{array} & \begin{bmatrix} 1&0&0&0&0\\ 0&1&0&1&1\\ 1&1&1&0&0\\ 0&0&1&1&1 \end{bmatrix}\end{array}$$

14.



14.
$$\begin{array}{c|c} & e_1\ e_2\ e_3\ e_4\ e_5\ e_6\ e_7\ e_8 \\ \begin{array}{c} a\\ b\\ c\\ d\end{array} & \begin{bmatrix} 1&1&1&1&0&0&0&0\\ 1&1&1&0&1&0&0&0\\ 0&0&0&0&1&1&1&1\\ 0&0&0&1&0&1&1&1 \end{bmatrix}\end{array}$$

15.

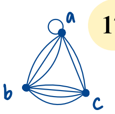


$$\begin{array}{c|c} & a\ b\ c\ d \\ \begin{array}{c} a\\ b\\ c\\ d\end{array} & \begin{bmatrix} 1&0&2&1\\ 0&1&1&2\\ 2&1&1&0\\ 1&2&0&1 \end{bmatrix}\end{array}$$

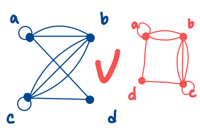
15.
$$\begin{array}{c|c} & e_1\ e_2\ e_3\ e_4\ e_5\ e_6\ e_7\ e_8\ e_9\ e_{10} \\ \begin{array}{c} a\\ b\\ c\\ d\end{array} & \begin{bmatrix} 1&0&0&0&1&1&0&0&1&0\\ 0&1&0&0&0&0&1&1&0&1\\ 0&0&1&0&1&1&0&0&0&1\\ 0&0&0&1&0&0&1&1&1&0 \end{bmatrix}\end{array}$$

In Exercises 16–18 draw an undirected graph represented by the given adjacency matrix.

16.
$$\begin{array}{c|c} & a\ \ b\ \ c \\ \begin{array}{c} a\\ b\\ c\end{array} & \begin{bmatrix} 1&3&2\\ 3&0&4\\ 2&4&0 \end{bmatrix}\end{array}$$



17.
$$\begin{array}{c|c} & a\ \ b\ \ c\ \ d \\ \begin{array}{c} a\\ b\\ c\\ d\end{array} & \begin{bmatrix} 1&2&0&1\\ 2&0&3&0\\ 0&3&1&1\\ 1&0&1&0 \end{bmatrix}\end{array}$$

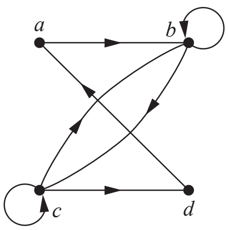


18.
$$\begin{bmatrix} 0&1&3&0&4\\ 1&2&1&3&0\\ 3&1&1&0&1\\ 0&3&0&0&2\\ 4&0&1&2&3 \end{bmatrix}$$

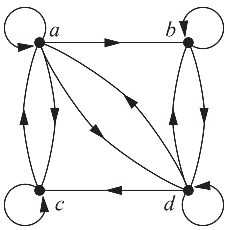
In Exercises 19–21 find the adjacency matrix of the given directed multigraph with respect to the vertices listed in alphabetic order.

19.

$$\begin{array}{c|c} & \begin{matrix} a & b & c & d \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

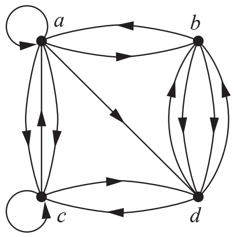


20.



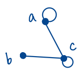
21.

$$\begin{array}{c|c} & \begin{matrix} a & b & c & d \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \end{matrix} & \begin{bmatrix} 1 & 1 & 2 & 1 \\ 1 & 0 & 0 & 2 \\ 1 & 0 & 1 & 1 \\ 0 & 2 & 1 & 0 \end{bmatrix} \end{array}$$

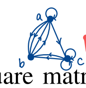


In Exercises 22–24 draw the graph represented by the given adjacency matrix.

22. $\begin{array}{c|c} & \begin{matrix} a & b & c \end{matrix} \\ \begin{matrix} a \\ b \\ c \end{matrix} & \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \end{array}$



23. $\begin{array}{c|c} & \begin{matrix} a & b & c \end{matrix} \\ \begin{matrix} a \\ b \\ c \end{matrix} & \begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 0 \\ 0 & 2 & 2 \end{bmatrix} \end{array}$



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

25. Is every zero–one square matrix that is symmetric and has zeros on the diagonal the adjacency matrix of a simple graph? yes ?

26. Use an incidence matrix to represent the graphs in Exercises 1 and 2.

27. Use an incidence matrix to represent the graphs in Exercises 13–15.

***28.** What is the sum of the entries in a row of the adjacency matrix for an undirected graph? For a directed graph?
degree of the vertex
out-degree for the vertex

***29.** What is the sum of the entries in a column of the adjacency matrix for an undirected graph? For a directed graph?

30. What is the sum of the entries in a row of the incidence matrix for an undirected graph?

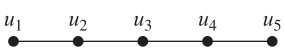
31. What is the sum of the entries in a column of the incidence matrix for an undirected graph?

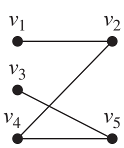
***32.** Find an adjacency matrix for each of these graphs.
a) K_n b) C_n c) W_n d) $K_{m,n}$ e) Q_n

***33.** Find incidence matrices for the graphs in parts (a)–(d) of Exercise 32.

In Exercises 34–44 determine whether the given pair of graphs is isomorphic. Exhibit an isomorphism or provide a rigorous argument that none exists.

34.



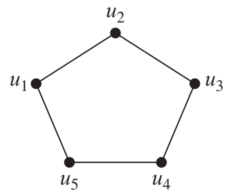


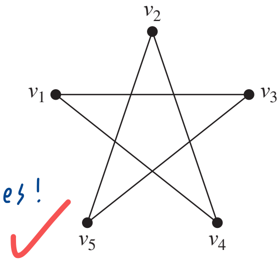
yes !

u	1	2	3	4	5
f(u)	1	2	5	4	3

35.

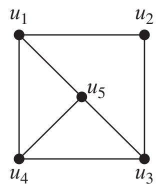
u	1	2	3	4	5
f(u)	1	3	5	2	4

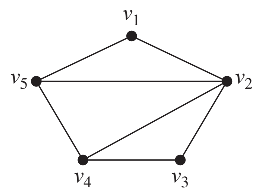




yes !

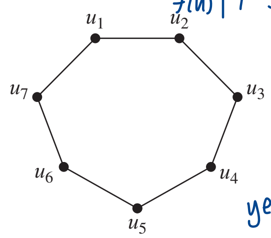
36.

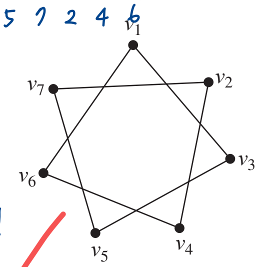




u	1	2	3	4	5	6	7
f(u)	1	3	5	7	2	4	6

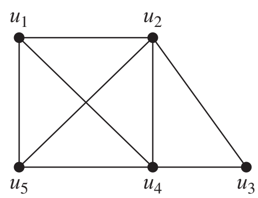
37.

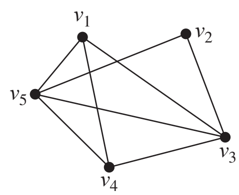




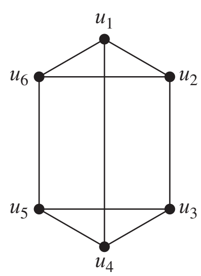
yes !

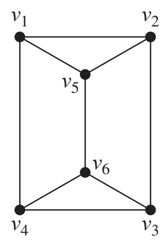
38.



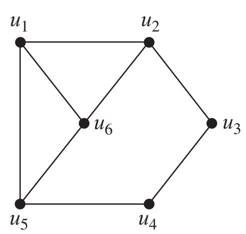


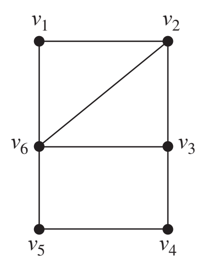
39.

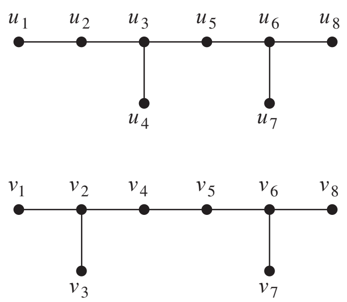


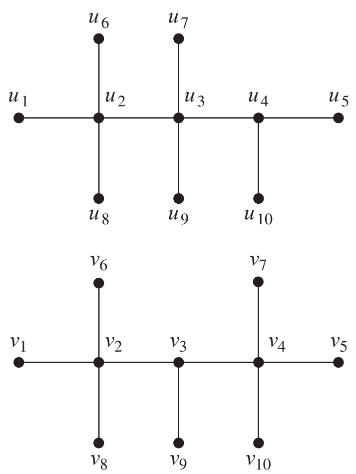


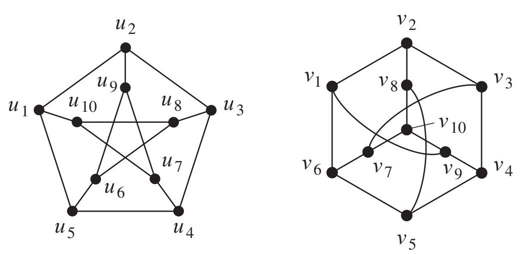
40.

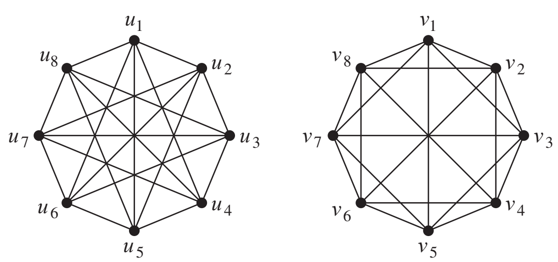




41. 

42. 

43. 

44. 

45. Show that isomorphism of simple graphs is an equivalence relation.

46. Suppose that G and H are isomorphic simple graphs. Show that their complementary graphs $\overline{G}$ and $\overline{H}$ are also isomorphic.

47. Describe the row and column of an adjacency matrix of a graph corresponding to an isolated vertex.

48. Describe the row of an incidence matrix of a graph corresponding to an isolated vertex.

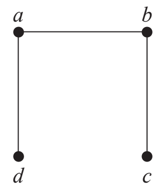
49. Show that the vertices of a bipartite graph with two or more vertices can be ordered so that its adjacency matrix has the form

$$\begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{B} & \mathbf{0} \end{bmatrix},$$

where the four entries shown are rectangular blocks.

A simple graph G is called **self-complementary** if G and $\overline{G}$ are isomorphic.

50. Show that this graph is self-complementary.



51. Find a self-complementary simple graph with five vertices.

***52.** Show that if G is a self-complementary simple graph with v vertices, then $v \equiv 0$ or $1 \pmod 4$.

53. For which integers n is C_n self-complementary?

54. How many nonisomorphic simple graphs are there with n vertices, when n is

a) 2? b) 3? c) 4?

55. How many nonisomorphic simple graphs are there with five vertices and three edges?

56. How many nonisomorphic simple graphs are there with six vertices and four edges?

57. Are the simple graphs with the following adjacency matrices isomorphic?

a) $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$ yes !

b) $\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$ no !

c) $\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$ no !

58. Determine whether the graphs without loops with these incidence matrices are isomorphic.

a) $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$

b) $\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$

The multigraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if there exists a bijection with the property that, if a has a loop, f(a) has one, if a has multiple edges to b, f(a) has the same amount to f(b)..

59. Extend the definition of isomorphism of simple graphs to undirected graphs containing loops and multiple edges.

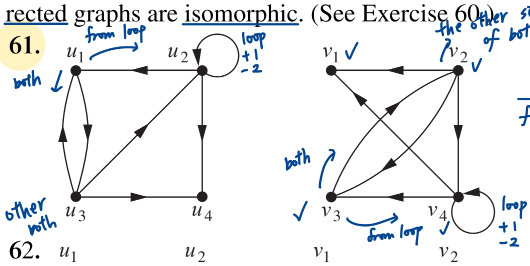
60. Define isomorphism of directed graphs.

if a has an edge from a to b, then f(a) has an edge from f(a) to f(b).

59.
$G = (V_1, E_1)$ is isomorphic to $H = (V_2, E_2)$ iff there exist function f from V_1 to V_2 and g from E_1 to E_2 such that each is a one-to-one correspondence and for every edge e in E_1 the endpoints of g(e) are f(v) and f(w) where v and w are the endpoints of e.

In Exercises 61–64 determine whether the given pair of directed graphs are isomorphic. (See Exercise 60.)

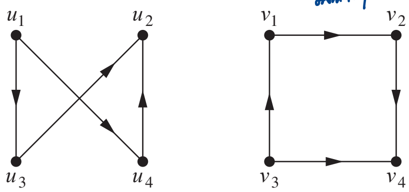
61.



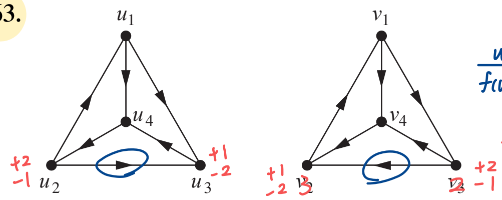
u	1	2	3	4
f(u)	3	4	2	1

yes!

62.



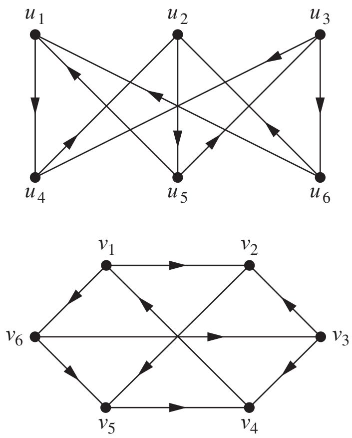
63.



u	1	2	3	4
f(u)	1	3	2	4

~~no!~~ yes

64.



65. Show that if G and H are isomorphic directed graphs, then the converses of G and H (defined in the preamble of Exercise 67 of Section 10.2) are also isomorphic.

66. Show that the property that a graph is bipartite is an isomorphic invariant.

67. Find a pair of nonisomorphic graphs with the same degree sequence (defined in the preamble to Exercise 36 in Section 10.2) such that one graph is bipartite, but the other graph is not bipartite.

***68.** How many nonisomorphic directed simple graphs are there with n vertices, when n is

a) 2? **b)** 3? **c)** 4?

***69.** What is the product of the incidence matrix and its transpose for an undirected graph?

***70.** How much storage is needed to represent a simple graph with n vertices and m edges using

a) adjacency lists?

b) an adjacency matrix?

c) an incidence matrix?

A **devil's pair** for a purported isomorphism test is a pair of nonisomorphic graphs that the test fails to show that they are not isomorphic.

71. Find a devil's pair for the test that checks the degree sequence (defined in the preamble to Exercise 36 in Section 10.2) in two graphs to make sure they agree.

72. Suppose that the function f from V_1 to V_2 is an isomorphism of the graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Show that it is possible to verify this fact in time polynomial in terms of the number of vertices of the graph, in terms of the number of comparisons needed.

10.4 Connectivity

Introduction

Many problems can be modeled with paths formed by traveling along the edges of graphs. For instance, the problem of determining whether a message can be sent between two computers using intermediate links can be studied with a graph model. Problems of efficiently planning routes for mail delivery, garbage pickup, diagnostics in computer networks, and so on can be solved using models that involve paths in graphs.

Paths

Informally, a **path** is a sequence of edges that begins at a vertex of a graph and travels from vertex to vertex along edges of the graph. As the path travels along its edges, it visits the vertices along this path, that is, the endpoints of these edges.