Chapter 10 Graph

Problems in almost every conceivable discipline can be solved using graph models.

10.1 Graphs and Graph Models

Definition 1 : What is `Graph` ?

A graph G = (V, E) consists of V, **a nonempty set of vertices** (or nodes) and E, **a set of edges**. Each edge has either one or two vertices associated with it, called its endpoints. An edge is said to connect its endpoints.

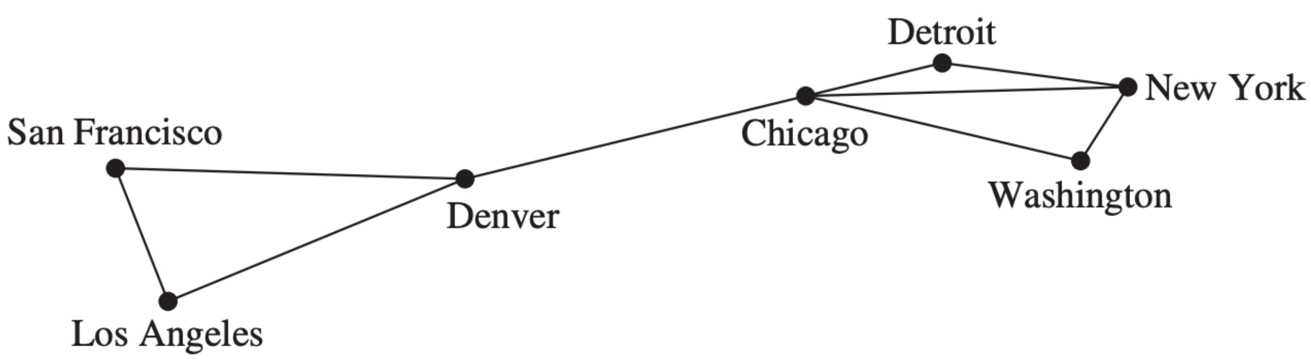
Remark: The set of vertices V of a graph G may be infinite. A graph with an infinite vertex set or an infinite number of edges is called an infinite graph, and in comparison, a graph with a finite vertex set and a finite edge set is called a finite graph. In this book we will usually consider only finite graphs.

the way we draw a graph is arbitrary, as long as the correct connections between vertices are depicted. So crossing and curve lines are all OK

What is a `simple Graph` ?

A graph in which each edge connects two different vertices and where **no two edges connect the same pair of vertices** is called a simple graph

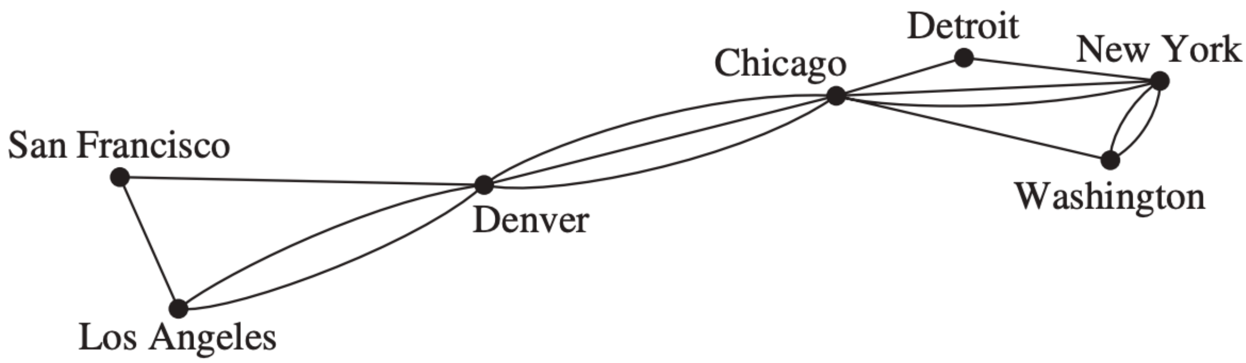
when there is an edge of a simple graph associated to {u, v}, we can also say, without possible confusion, that {u, v} is an edge of the graph.



What is a `Multigraph`

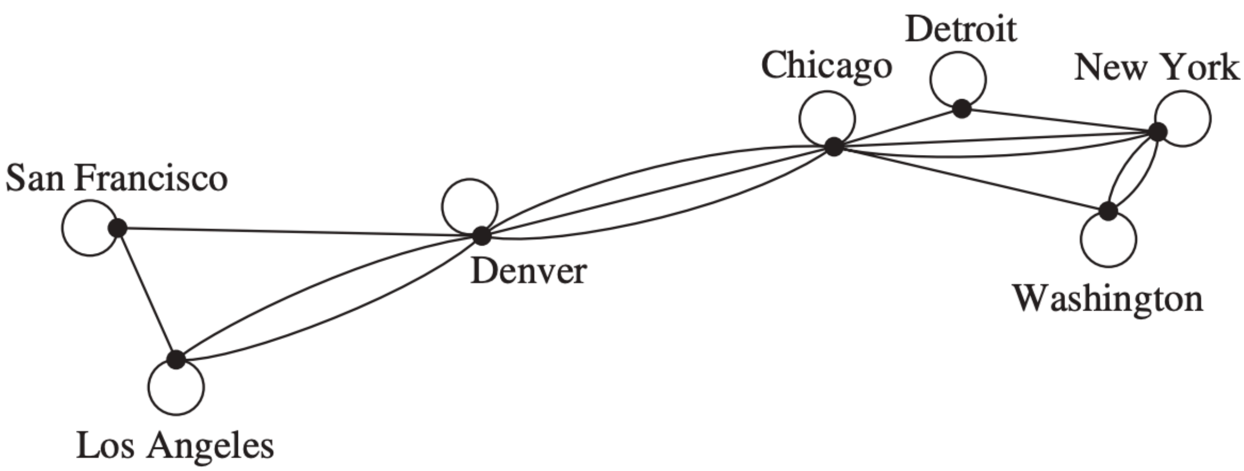
Graphs that may have **multiple edges connecting the same vertices** are called multigraphs

When there are m different edges associated to the same unordered pair of vertices {u, v}, we also say that {u, v} is an edge of multiplicity m. That is, we can think of this set of edges as m different copies of an edge {u, v}.



What is a `Pseudograph`

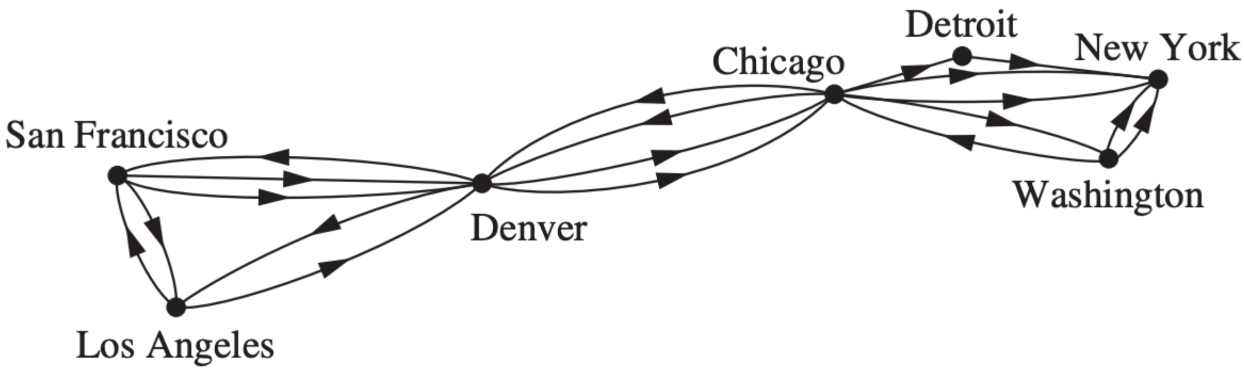
Graphs that may **include loops**, and possibly **multiple edges** connecting the same pair of vertices or a vertex to itself, are sometimes called pseudographs.



Definition 2 : What is a `Digraph`

A directed graph (or digraph) (V, E) consists of a nonempty set of vertices V and a set of directed edges (or arcs) E. Each directed edge is associated with an ordered pair of vertices. The directed edge associated with the ordered pair (u, v) is said to start at u and end at v.

When a directed graph has no loops and has no multiple directed edges, it is called a **simple directed graph**



What is a `mixed Graph`

A graph with **both directed and undirected edges** is called a mixed graph

TABLE 1 **Graph Terminology.**			
Type	***Edges***	***Multiple Edges Allowed?***	***Loops Allowed?***
Simple graph	Undirected	No	No
Multigraph	Undirected	Yes	No
Pseudograph	Undirected	Yes	Yes
Simple directed graph	Directed	No	No
Directed multigraph	Directed	Yes	Yes
Mixed graph	Directed and undirected	Yes	Yes

NOTE : loops are allowed in `directed multigraph` , while not allowed in `multigraph`

- Are the edges of the graph undirected or directed (or both)?
- If the graph is undirected, are multiple edges present that connect the same pair of vertices? If the graph is directed, are multiple directed edges present?
- Are loops present?

Answering such questions helps us understand graphs. It is less important to remember the particular terminology used.

10.1.1 Graph Models

When we build a graph model, we need to make sure that we have correctly answered the three key questions we posed about the structure of a graph

10.2 Graph Terminology and Special Types of Graphs

10.2.1 Basic Terminology

UNDIRECTED GRAPH

Definition 1 : What is an `adjacent` ?

Two vertices u and v in an undirected graph G are called adjacent (or neighbors) in G if u and v are **endpoints of an edge e of G**. Such an edge e is called incident with the vertices u and v and e is said to connect u and v.

Definition 2 : What is a `neighborhood` ?

The set of all neighbors of a vertex v of G = (V, E), denoted by N(v), is called the neighborhood of v. If A is a subset of V, we denote by N(A) the set of all vertices in G that are adjacent to at least one vertex in A. So, $\cup_{u\in A} N(u)$.

Definition 3 : What is `degree` ?

The degree of a vertex in an undirected graph is **the number of edges incident with it**, except that a loop at a vertex contributes twice to the degree of that vertex. The degree of the vertex v is denoted by deg(v).

EXAMPLE 1 What are the degrees and what are the neighborhoods of the vertices in the graphs G and H displayed in Figure 1?

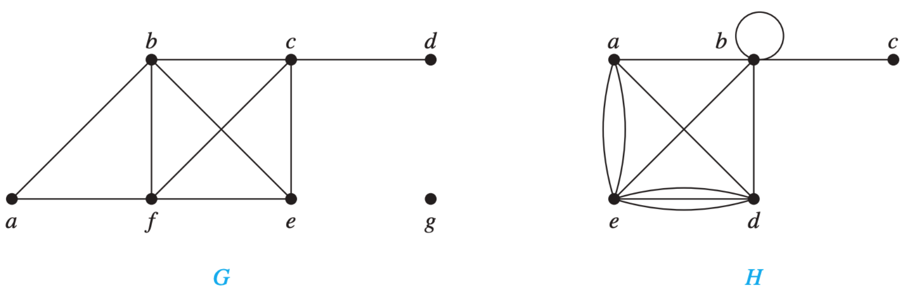


FIGURE 1 **The undirected graphs *G* and *H*.**

Solution: In G, $\deg(a) = 2$, $\deg(b) = \deg(c) = \deg(f) = 4$, $\deg(d) = 1$, $\deg(e) = 3$, and $\deg(g) = 0$. The neighborhoods of these vertices are $N(a) = \{b, f\}$, $N(b) = \{a, c, e, f\}$, $N(c) = \{b, d, e, f\}$, $N(d) = \{c\}$, $N(e) = \{b, c, f\}$, $N(f) = \{a, b, c, e\}$, and $N(g) = \emptyset$. In H, $\deg(a) = 4$, $\deg(b) = \deg(e) = 6$, $\deg(c) = 1$, and $\deg(d) = 5$. The neighborhoods of these vertices are $N(a) = \{b, d, e\}$, $N(b) = \{a, b, c, d, e\}$, $N(c) = \{b\}$, $N(d) = \{a, b, e\}$, and $N(e) = \{a, b, d\}$. ◀

Definition : What is `isolate` and `pendant` ?

A vertex of **degree zero is called isolated**. It follows that an isolated vertex is not adjacent to any vertex. Vertex g in graph G in Example 1 is isolated. **A vertex is pendant if and only if it has degree one**. Consequently, a pendant vertex is adjacent to exactly one other vertex. Vertex d in graph G in Example 1 is pendant.

Theorem 1 : Handshaking Theorem

Let G = (V, E) be an undirected graph with m edges. Then

$$2m = \sum_{v \in V} \deg(v)$$

(Note that this applies even if multiple edges and loops are present.)

Each edge contributes two to the sum of the degrees of the vertices because an edge is incident with exactly two (possibly equal) vertices. This means that the sum of the degrees of the vertices is twice the number of edges. We have the result in Theorem 1, which is sometimes called the handshaking theorem (and is also often known as the handshaking lemma), because of the analogy between an edge having two endpoints and a handshake involving two hands

EXAMPLE 3 How many edges are there in a graph with 10 vertices each of degree six?

Solution: Because the sum of the degrees of the vertices is $6 \cdot 10 = 60$, it follows that $2m = 60$ where m is the number of edges. Therefore, $m = 30$. ◀

Theorem 2 :

An undirected graph has an even number of vertices of odd degree.

Proof: Let V_1 and V_2 be the set of vertices of even degree and the set of vertices of odd degree, respectively, in an undirected graph $G = (V, E)$ with m edges. Then

$$2m = \sum_{v \in V} \deg(v) = \sum_{v \in V_1} \deg(v) + \sum_{v \in V_2} \deg(v).$$

Because $\deg(v)$ is even for $v \in V_1$, the first term in the right-hand side of the last equality is even. Furthermore, the sum of the two terms on the right-hand side of the last equality is even, because this sum is $2m$. Hence, the second term in the sum is also even. Because all the terms in this sum are odd, there must be an even number of such terms. Thus, there are an even number of vertices of odd degree. ◁

DIRECTED GRAPH

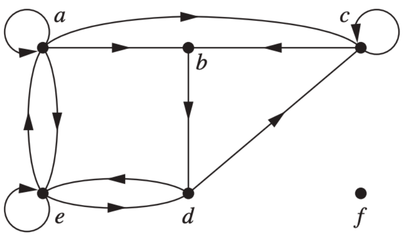
Definition 4 : What is `adjacent` in directed graph?

When (u, v) is an edge of the graph G with directed edges, u is said to be adjacent to v and v is said to be adjacent from u. The vertex u is called the initial vertex of (u, v), and v is called the terminal or end vertex of (u, v). The initial vertex and terminal vertex of a loop are the same.

Definition 5 : What is `degree` in directed graph?

In a graph with directed edges the in-degree of a vertex v, denoted by $\deg^-(v)$, is the number of edges with v as their terminal vertex. The out-degree of v, denoted by deg+(v), is the number of edges with v as their initial vertex. (Note that a loop at a vertex contributes 1 to both the in-degree and the out-degree of this vertex.)

EXAMPLE 4 Find the in-degree and out-degree of each vertex in the graph G with directed edges shown in Figure 2.



G

Solution: The in-degrees in G are $\deg^-(a) = 2$, $\deg^-(b) = 2$, $\deg^-(c) = 3$, $\deg^-(d) = 2$, $\deg^-(e) = 3$, and $\deg^-(f) = 0$. The out-degrees are $\deg^+(a) = 4$, $\deg^+(b) = 1$, $\deg^+(c) = 2$, $\deg^+(d) = 2$, $\deg^+(e) = 3$, and $\deg^+(f) = 0$. ◀

Theorem 3 :

Let G = (V, E) be a graph with directed edges. Then

$$\sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v) = |E|$$

NOTE: the theorem 1 is still true here

Definition : What is an `underlying undirected graph` ?

There are many properties of a graph with directed edges that do not depend on the direction of its edges. Consequently, it is often useful to ignore these directions. **The undirected graph that results from ignoring directions of edges is called the `underlying undirected graph`.** A graph with directed edges and its underlying undirected graph have the same number of edges.

10.2.3 Some Simple Special Graphs

Complete Graphs

EXAMPLE 5 **Complete Graphs** A **complete graph on n vertices**, denoted by K_n, is a simple graph that contains exactly one edge between each pair of distinct vertices. The graphs K_n, for $n = 1, 2, 3, 4, 5, 6$, are displayed in Figure 3. A simple graph for which there is at least one pair of distinct vertex not connected by an edge is called **noncomplete**. ◀

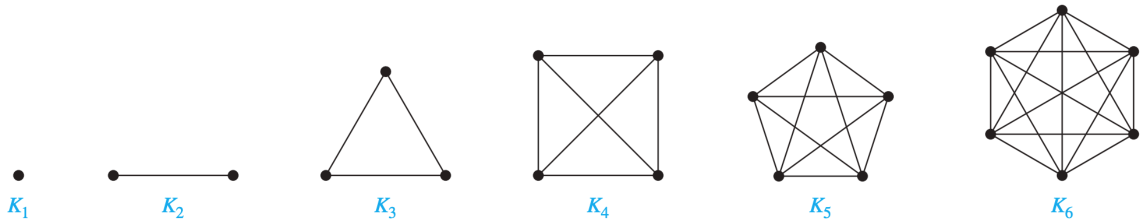


FIGURE 3 **The graphs K_n for $1 \leq n \leq 6$.**

Cycles

EXAMPLE 6 **Cycles** A **cycle C_n**, $n \geq 3$, consists of n vertices $v_1, v_2, \ldots, v_n$ and edges $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}$, and $\{v_n, v_1\}$. The cycles C_3, C_4, C_5, and C_6 are displayed in Figure 4. ◀

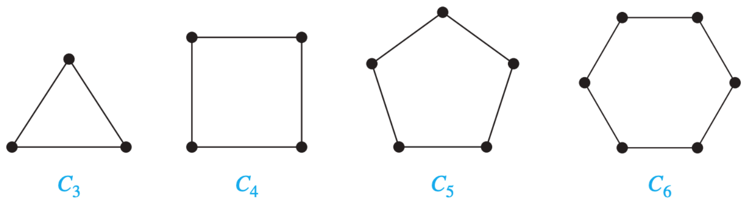


FIGURE 4 **The cycles C_3, C_4, C_5, and C_6.**

Wheels (Cycles with a 'center')

EXAMPLE 7 **Wheels** We obtain a **wheel W_n** when we add an additional vertex to a cycle C_n, for $n \geq 3$, and connect this new vertex to each of the n vertices in C_n, by new edges. The wheels W_3, W_4, W_5, and W_6 are displayed in Figure 5. ◀

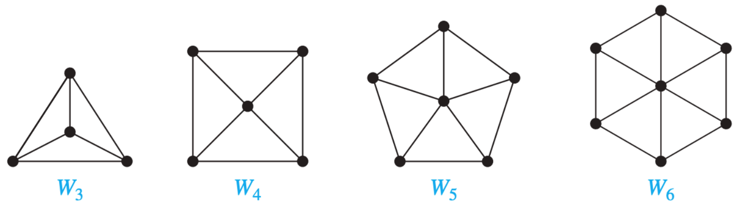


FIGURE 5 **The wheels W_3, W_4, W_5, and W_6.**

n-Cubes

EXAMPLE 8 ***n*-Cubes** An ***n*-dimensional hypercube**, or ***n*-cube**, denoted by Q_n, is a graph that has vertices representing the 2^n bit strings of length n. Two vertices are adjacent if and only if the bit strings that they represent differ in exactly one bit position. We display Q_1, Q_2, and Q_3 in Figure 6.

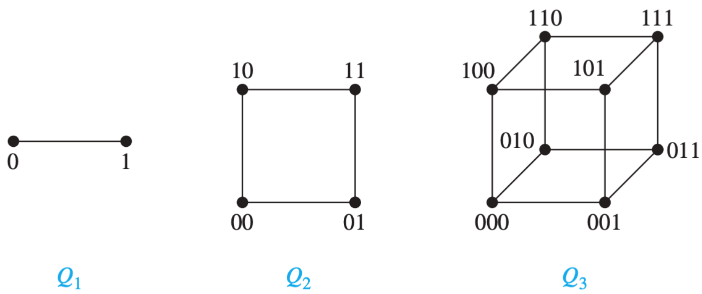


FIGURE 6 **The *n*-cube Q_n, $n = 1, 2, 3$.**

Note that you can construct the (n + 1)-cube Qn+1 from the n-cube Qn by making two copies of Qn, prefacing the labels on the vertices with a 0 in one copy of Qn and with a 1 in the other copy of Qn, and adding edges connecting two vertices that have labels differing only in the first bit.

D6 : what is a `partite graph` ?

A simple graph G is called bipartite if its vertex set V can be partitioned into two disjoint sets V1 and V2 such that every edge in the graph connects a vertex in V1 and a vertex in V2 (so that no edge in G connects either two vertices in V1 or two vertices in V2). When this condition holds, we call the pair (V1, V2) a bipartition of the vertex set V of G.

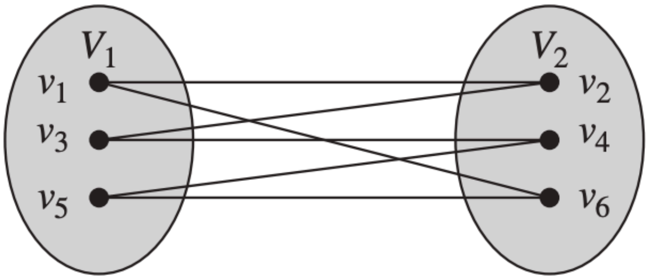


FIGURE 7 Showing that C_6 is bipartite.

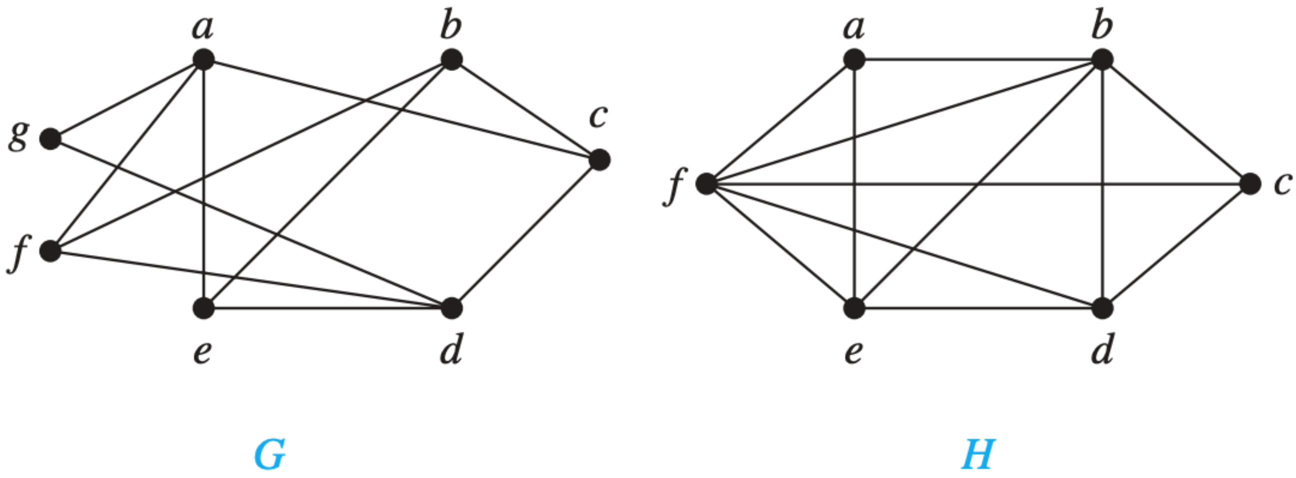


FIGURE 8 The undirected graphs G and H.

Solution: Graph G is bipartite because its vertex set is the union of two disjoint sets, $\{a, b, d\}$ and $\{c, e, f, g\}$, and each edge connects a vertex in one of these subsets to a vertex in the other subset. (Note that for G to be bipartite it is not necessary that every vertex in $\{a, b, d\}$ be adjacent to every vertex in $\{c, e, f, g\}$. For instance, b and g are not adjacent.)

Graph H is not bipartite because its vertex set cannot be partitioned into two subsets so that edges do not connect two vertices from the same subset. (The reader should verify this by considering the vertices a, b, and f.) ◀

Theorem 4 : dermine whether a graph is a bipartite graph

A simple graph is bipartite if and only if it is possible to assign one of two different colors to each vertex of the graph so that no two adjacent vertices are assigned the same color.

EXAMPLE 13 **Complete Bipartite Graphs** A **complete bipartite graph** $K_{m,n}$ is a graph that has its vertex set partitioned into two subsets of m and n vertices, respectively with an edge between two vertices if and only if one vertex is in the first subset and the other vertex is in the second subset. The complete bipartite graphs $K_{2,3}$, $K_{3,3}$, $K_{3,5}$, and $K_{2,6}$ are displayed in Figure 9. ◀

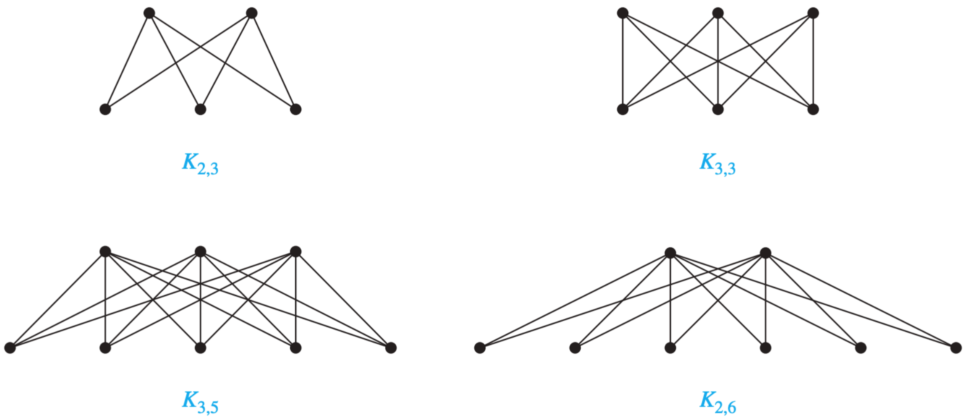


FIGURE 9 **Some complete bipartite graphs.**

Definition : What is a `matching` ?

a matching M in a simple graph G = (V, E) is a subset of the set E of edges of the graph such that no two edges are incident with the same vertex. In other words, a matching is a subset of edges such that if {s, t} and {u, v} are distinct edges of the matching,then s, t, u, and v are distinct.

A vertex that is the endpoint of an edge of a matching M is said to be **matched** in M; otherwise it is said to be **unmatched**. A **maximum matching** is a matching with the largest number of edges.

We say that a matching M in a bipartite graph G = (V, E) with bipartition (V_1, V_2) is a **complete matching** from V1 to V2 if every vertex in V1 is the endpoint of an edge in the matching, or equivalently, if |M| = $|V_1|$

NOTE : for complete matching, V1 should have fewer vertices than V2 because only under this condition, there are sufficient vertices to match.

Theorem 5 : HALL'S MARRIAGE THEOREM

The bipartite graph G = (V, E) with bipartition (V1, V2) has a **complete matching** from V1 to V2 if and only if |N(A)| ≥ |A| for **all subsets A of V1**.

NOTE : N(A) must be a subset of V2 for there are no edges between vertices in V1

Proof: We first prove the *only if* part of the theorem. To do so, suppose that there is a complete matching M from V_1 to V_2. Then, if $A \subseteq V_1$, for every vertex $v \in A$, there is an edge in M connecting v to a vertex in V_2. Consequently, there are at least as many vertices in V_2 that are neighbors of vertices in V_1 as there are vertices in V_1. It follows that $|N(A)| \geq |A|$.

> to prove the IF part of the theorem, we use **strong induction**
>
> Some vivid proving

This enlight me on our daily promble-solving questions, when we resort to computer to solve problems, make the algorithm close to the essence as much as we can, don't just simulate the process.

10.2.7 New Graphs from old

> ## Definition 7 : What is a `subgraph` ?
>
> ---
>
> A subgraph of a graph G = (V, E) is a graph H = (W, F), where W ⊆ V and F ⊆ E. A subgraph H of G is a proper subgraph of G if H ≠ G.

When edges and vertices are removed from a graph, without removing endpoints of any remaining edges, a smaller graph is obtained. Such a graph is called a subgraph of the original graph.

> ## D8 : What is a `subgraph induce` ?
>
> ---
>
> Let G = (V, E) be a simple graph. The subgraph induced by a subset W of the vertex set V is the graph (W, F), where **the edge set F contains an edge in E if and only if both endpoints of this edge are in W**

we can also remove edges to obtain a subgraph

- add edge e(not subgraph)

$$G + e = G(V, \{e\} \bigcup E)$$

- remove edge e

$$G - e = G(V, E - \{e\})$$

> ## D : What is a `graph contraction` ?
>
> ---
>
> edge contraction, which removes an edge e with endpoints u and v and merges u and v into a new single vertex w, and for each edge with u or v as an endpoint replaces the edge with one with w as endpoint in place of u or v and with the same second endpoint
>
> ***NOTE : the new graph is NOT a subgraph of the original one***

error at page 698 : w should be v in this place

n, which removes an edge e with endpoints u and v and merges u

and w into a new single vertex w, and for each edge with u or
with one with w as endpoint in place of u or v and with the
contraction of the edge e with endpoints u and v in the graph
$G' = (V', E')$ (which is not a subgraph of G), where $V' = V$ -

remove vertex:

- When we remove a vertex v and all edges incident to it from G = (V, E), we produce a subgraph, denoted by G − v

D9 : What is `graph union` ?

The union of two **simple graphs** G1 = (V1, E1) and G2 = (V2, E2) is the simple graph with vertex set V1 ∪ V2 and edge set E1 ∪ E2. The union of G1 and G2 is denoted by G1 ∪ G2.

EXAMPLE 20 Find the union of the graphs G_1 and G_2 shown in Figure 17(a).

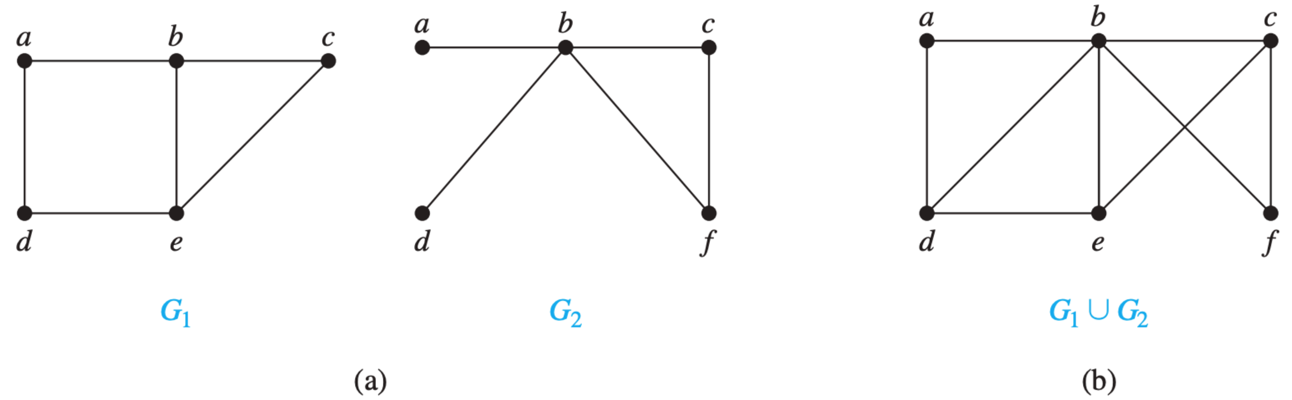


FIGURE 17 **(a) The simple graphs G_1 and G_2. (b) Their union $G_1 \cup G_2$.**

10.3 Representing Graphs and Graph Isomorphism

Introduction

- there are many ways to represent a graph, choose the most appropriate one is important.

10.3.2 Representing Graphs

1. Adjacency list

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. ◀

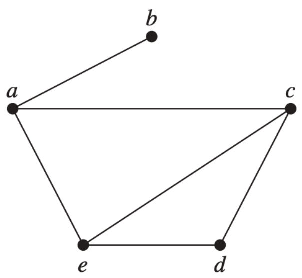


FIGURE 1 **A simple graph.**

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

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

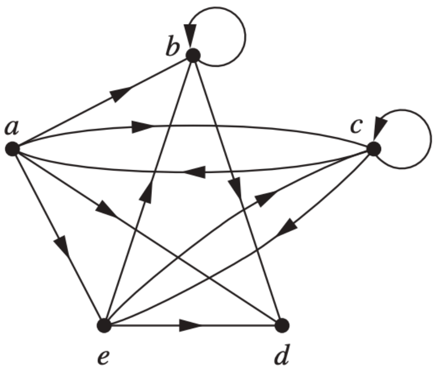


FIGURE 2 **A directed graph.**

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

2. adjacency matrices

> The adjacency matrix A (or 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 vi and vj are adjacent, and 0 as its (i, j)th entry when they are not adjacent.
>
> $$a_{ij} = \begin{cases} 1 & \text{if \{i,j\} is a edge of G} \\ 0 & \text{otherwise} \end{cases}$$
>
> *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.*

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

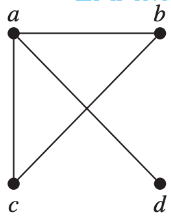


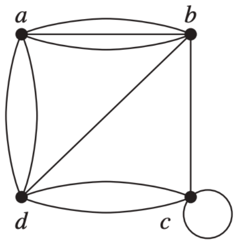
FIGURE 3
A simple graph.

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

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

$$a_{ij} = \begin{cases} n & \text{if there are n edges \{i,j\}} \\ 0 & \text{otherwise} \end{cases}$$

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

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

Adjacency matrices can also be used to represent directed multigraphs

$$a_{ij} = \begin{cases} 1 & \text{if } v_i \to v_j \\ 0 & \text{otherwise} \end{cases}$$

- When there are few edges in the graph, use adjacency list
- When there are many edges in the graph, use adjacency matrix

3. Incidence matrix

> Let G = (V, E) be an undirected graph. Suppose that v1, v2, … , vn are the vertices and e1, e2, … , em are the edges of G. Then the incidence matrix with respect to this ordering of V and E is the n × m matrix M = $[m_{ij}]$, where
>
> $$m_{ij} = \begin{cases} 1 & \text{vertex i incide with edge j} \\ 0 & \text{otherwise} \end{cases}$$

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

Solution: The incidence matrix is

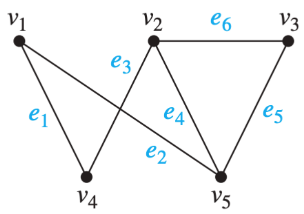


FIGURE 6 An undirected graph.

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

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

Solution: The incidence matrix for this graph is

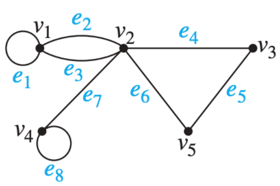


FIGURE 7 A pseudograph.

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

10.3.5 Isomorphism of Graphs

D1 : `isomorphism`

The simple graphs G1 = (V1, E1) and G2 = (V2, E2) are isomorphic if there exists a one-to-one and onto function f from V1 to V2 with the property that a and b are adjacent in G1 if and only if f(a) and f(b) are adjacent in G2, for all a and b in V1. 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.

MyUnderstanding: if (v_1, v_2) is edge in G1, then $(f(v_1), f(v_2))$ is still edge in G2, for all v_1, v_2

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. ◀

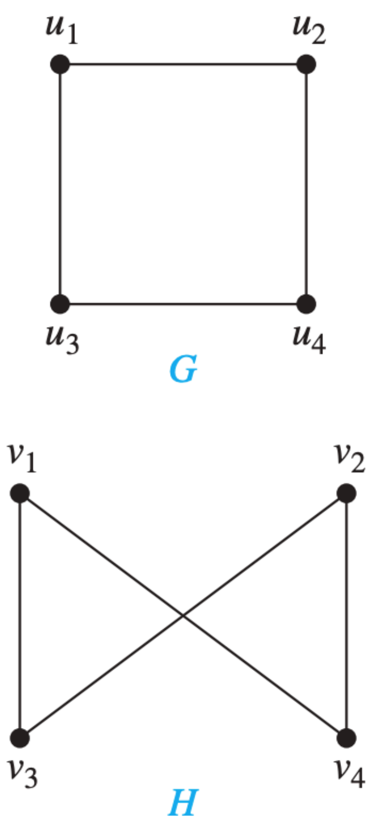


FIGURE 8 **The graphs G and H.**

10.3.6 Determining whether Two Simple Graphs are Isomorphic

MyU : This related to determine whether two simple graphs have the 'same' structure

Definition : `graph invariant`

A property preserved by isomorphism of graphs is called a graph invariant. For instance,

- the same number of vertices
- the same number of edges
- the same number of degree of each vertex
- the same Circuits

- ......

BUT There are no useful sets of invariants currently known that can be used to determine whether simple graphs are isomorphic.

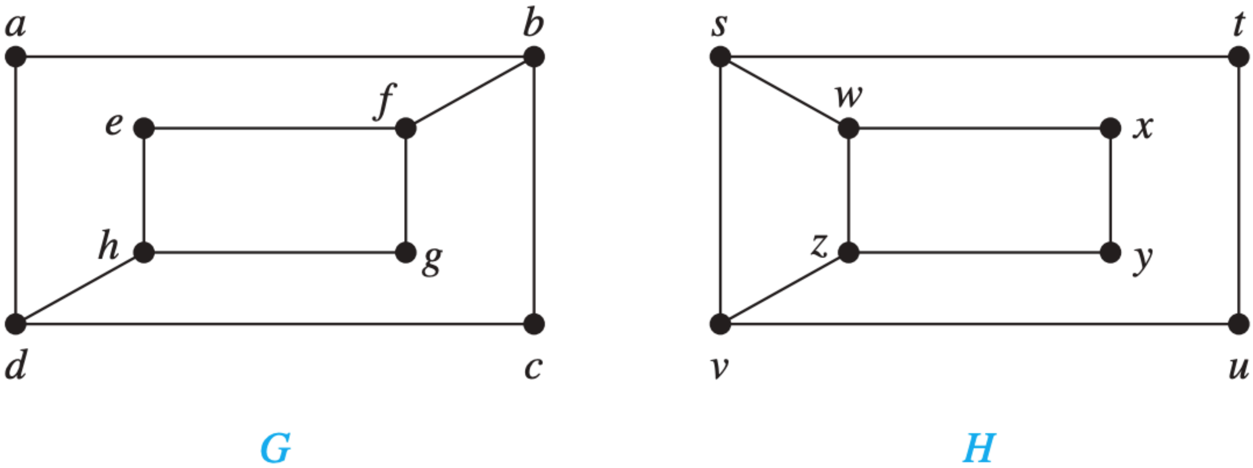
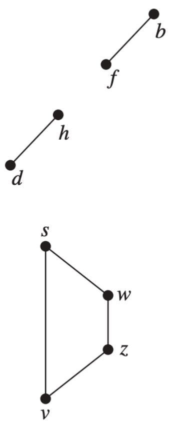


FIGURE 10 **The graphs *G* and *H*.**



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. ◀

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.

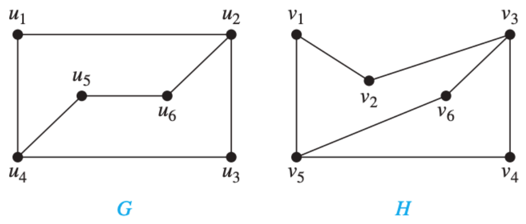


FIGURE 12 **Graphs G and H.**

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,

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

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. ◀

10.4 Connectivity

10.4.2 Paths

D1 : what is a `path`?

Let n be a nonnegative integer and G an undirected graph. A **path** of length n from u to v in G is a sequence of n edges e1, … , en of G for which there exists a sequence $x_0 = u, x_1, \ldots, x_{n-1}, x_n = v$ of vertices such that e_i has, for i = 1, … , n, the endpoints xi−1 and xi . When the graph is simple, we denote this path by its vertex sequence x0, x1, … , xn (because listing these vertices uniquely determines the path). The path is a **circuit** if it begins and ends at the same vertex, that is, if u = v, and has length greater than zero. The path or circuit is said to pass through the vertices x1, x2, … , xn−1 or traverse the edges e1, e2, … , en. A **path or circuit is simple** if it does not contain the same edge more than once.

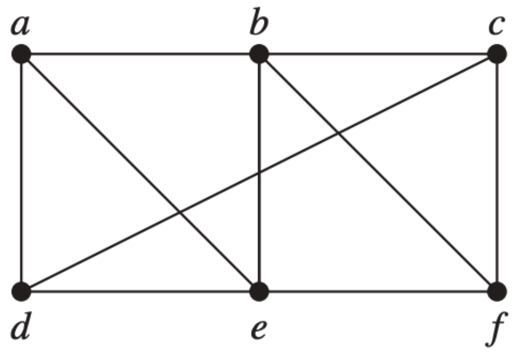


FIGURE 1 **A simple graph.**

EXAMPLE 1 In the simple graph shown in Figure 1, a, d, c, f, e is a simple path of length 4, because $\{a, d\}$, $\{d, c\}$, $\{c, f\}$, and $\{f, e\}$ are all edges. However, d, e, c, a is not a path, because $\{e, c\}$ is not an edge. Note that b, c, f, e, b is a circuit of length 4 because $\{b, c\}$, $\{c, f\}$, $\{f, e\}$, and $\{e, b\}$ are edges, and this path begins and ends at b. The path a, b, e, d, a, b, which is of length 5, is not simple because it contains the edge $\{a, b\}$ twice. ◀

D2 : what is a `path` in directed graph

Let n be a nonnegative integer and G a directed graph. A path of length n from u to v in G is a sequence of edges e1, e2, … , en of G such that e1 is associated with (x0, x1), e2 is associated with (x1, x2), and so on, with en associated with (xn−1, xn), where x0 = u and xn = v. When there are no multiple edges in the directed graph, this path is denoted by its vertex sequence x0, x1, x2, … , xn. A path of length greater than zero that begins and ends at the same vertex is called a circuit or cycle. A path or circuit is called simple if it does not contain the same edge more than once.

10.4.3 Connectedness in Undirected Graphs

D3: what is a `connected` graph?

An undirected graph is called connected if **there is a path between <u>every pair</u> of distinct vertices of the graph**. An undirected graph that is not connected is called disconnected. We say that we disconnect a graph when we remove vertices or edges, or both, to produce a disconnected subgraph.

Theorem 1 :

There is a simple path between every pair of distinct vertices of a connected undirected graph

Proof: Let u and v be two distinct vertices of the connected undirected graph $G = (V, E)$. Because G is connected, there is at least one path between u and v. Let $x_0, x_1, \ldots, x_n$, where $x_0 = u$ and $x_n = v$, be the vertex sequence of a path of least length. This path of least length is simple. To see this, suppose it is not simple. Then $x_i = x_j$ for some i and j with $0 \leq i < j$. This means that there is a path from u to v of shorter length with vertex sequence $x_0, x_1, \ldots, x_{i-1}, x_j, \ldots, x_n$ obtained by deleting the edges corresponding to the vertex sequence $x_i, \ldots, x_{j-1}$. ◁

Definition: what is a `connected component`

A connected component of a graph G is **a connected subgraph of G that is not a proper subgraph of another connected subgraph of G**. That is, a connected component of a graph G is a maximal connected subgraph of G. A graph G that is **not** connected has two or more connected components that are disjoint and have G as their union.

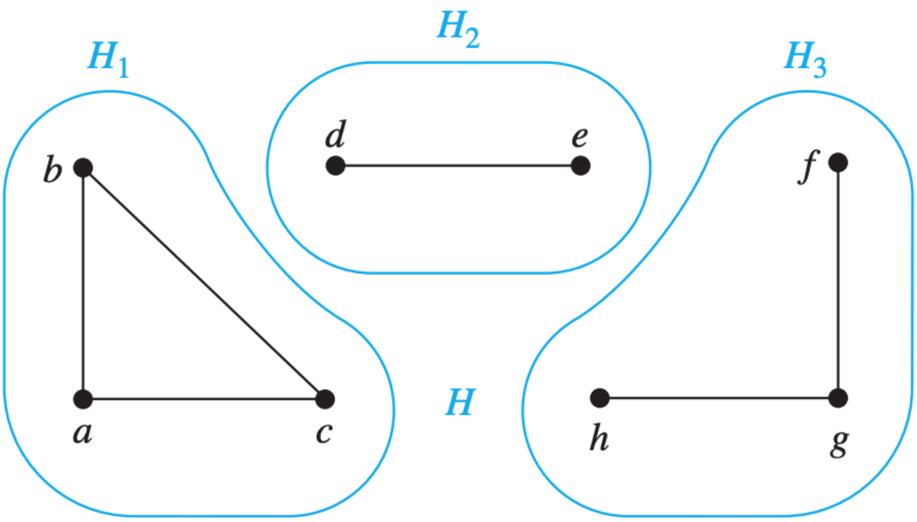


FIGURE 3 **The graph H and its connected components H_1, H_2, and H_3.**

EXAMPLE 5 What are the connected components of the graph H shown in Figure 3?

Solution: The graph H is the union of three disjoint connected subgraphs H_1, H_2, and H_3, shown in Figure 3. These three subgraphs are the connected components of H. ◀

10.4.4 How Connected is a Graph?

What is a `cut vertex` or `cut edge` ?

the removal from a graph of a vertex and all incident edges **produces a subgraph with more connected components**. Such vertices are called cut vertices(or articulation points).The removal of a cut vertex from a connected graph produces a subgraph that is not connected.

Analogously, an edge whose removal **produces a graph with more connected components than in the original graph** is called a cut edge or bridge.

EXAMPLE 7 Find the cut vertices and cut edges in the graph G_1 shown in Figure 4.

Solution: The cut vertices of G_1 are b, c, and e. The removal of one of these vertices (and its adjacent edges) disconnects the graph. The cut edges are $\{a, b\}$ and $\{c, e\}$. Removing either one of these edges disconnects G_1. ◀

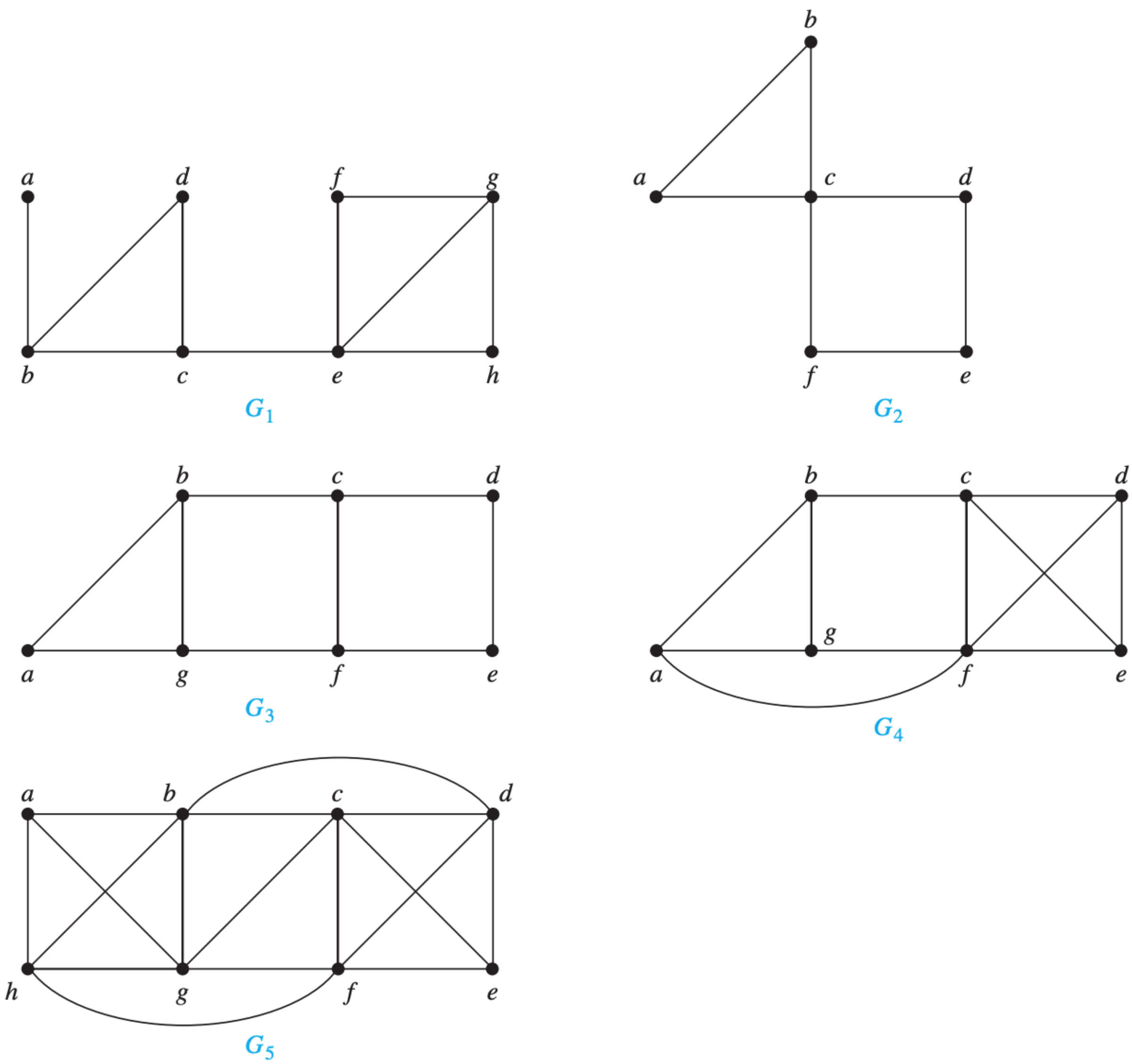


FIGURE 4 **Some connected graphs.**

what is the `vertex connectivity`

A subset V′ of the vertex set V of G = (V, E) is a **vertex cut**, or **separating set**, if G − V′ is disconnected.

We define the vertex connectivity of a noncomplete graph G, denoted by κ(G), as the **minimum number of vertices in a vertex cut**.

When G is a complete graph, it has no vertex cuts, because removing any subset of its vertices and all incident edges still leaves a complete graph. Consequently, we cannot define κ(G) as the minimum number of vertices in a vertex cut when G is complete. Instead, we set $\kappa(K_n) = n - 1$, the number of vertices needed to be removed to produce a graph with a single vertex.

EXAMPLE 8 Find the vertex connectivity for each of the graphs in Figure 4.

Solution: Each of the five graphs in Figure 4 is connected and has more than vertex, so each of these graphs has positive vertex connectivity. Because G_1 is a connected graph with a cut vertex, as shown in Example 7, we know that $\kappa(G_1) = 1$. Similarly, $\kappa(G_2) = 1$, because c is a cut vertex of G_2.

The reader should verify that G_3 has no cut vertices, but that $\{b, g\}$ is a vertex cut. Hence, $\kappa(G_3) = 2$. Similarly, because G_4 has a vertex cut of size two, $\{c, f\}$, but no cut vertices. It follows that $\kappa(G_4) = 2$. The reader can verify that G_5 has no vertex cut of size two, but $\{b, c, f\}$ is a vertex cut of G_5. Hence, $\kappa(G_5) = 3$. ◀

What is a `edge connectivity` in a graph?

The edge connectivity of a graph G. denoted by λ(G), is the minimum number of edges in an edge cut of G

Note that λ(G) = 0 if G is not connected. We also specify that λ(G) = 0 if G is a graph consisting of a single vertex. It follows that if G is a graph with n vertices, then $0 \leq \lambda(G) \leq n - 1$.

An inequalityality for vertex connectivity and edge connectivity

$$\kappa(G) \leq \lambda(G) \leq min_{v \in V} \deg(v)$$

10.4.5 Connectedness in Directed Graphs

D4 : `strong connected`

A directed graph is strongly connected if **there is a path from a to b and from b to a whenever a and b are vertices** in the graph.

D5 : `weak connected`

A directed graph is weakly connected if there is a path between every two vertices in the underlying undirected graph.

That is, a directed graph is weakly connected if and only if there is always a path between two vertices when the directions of the edges are disregarded. Clearly, any strongly connected directed graph is also weakly connected.

EXAMPLE 10 Are the directed graphs G and H shown in Figure 5 strongly connected? Are they weakly connected?

Solution: G is strongly connected because there is a path between any two vertices in this directed graph (the reader should verify this). Hence, G is also weakly connected. The graph H is not strongly connected. There is no directed path from a to b in this graph. However, H is weakly connected, because there is a path between any two vertices in the underlying undirected graph of H (the reader should verify this). ◀

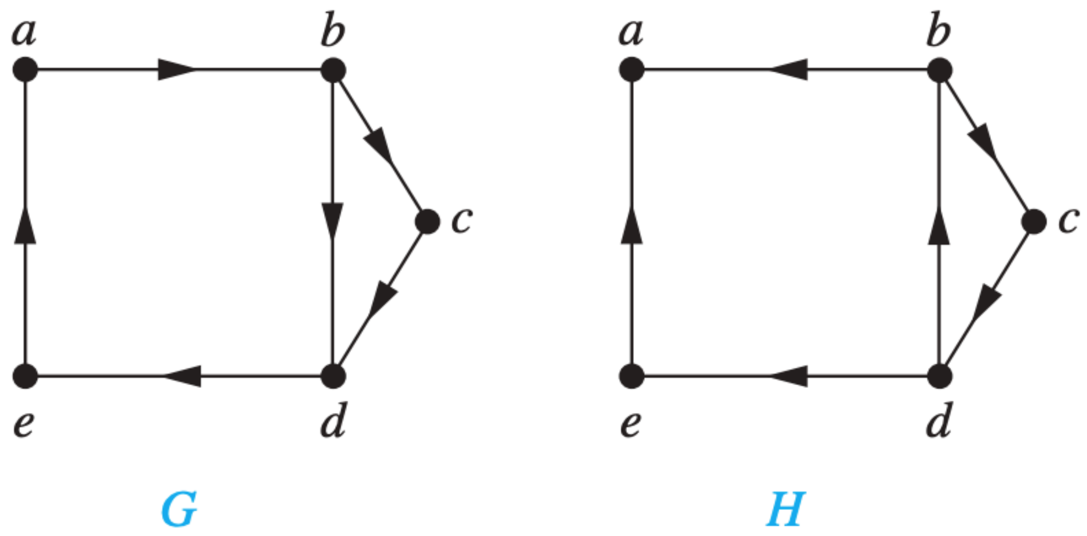


FIGURE 5 **The directed graphs G and H.**

D: strong component

The subgraphs of a directed graph G that are strongly connected but not contained in larger strongly connected subgraphs, that is, the maximal strongly connected subgraphs, are called the strongly connected components or strong components of G. Note that if a and b are two vertices in a directed graph, their strong components are either the same or disjoint.

EXAMPLE 11 The graph H in Figure 5 has three strongly connected components, consisting of the vertex a; the vertex e; and the subgraph consisting of the vertices b, c, and d and edges (b, c), (c, d), and (d, b). ◀

10.4.6 Paths and Isomorphism

Paths and circuits can be useful invariant to determine isomorphism

EXAMPLE 13 Determine whether the graphs G and H shown in Figure 6 are isomorphic.

Solution: Both G and H have six vertices and eight edges. Each has four vertices of degree three, and two vertices of degree two. So, the three invariants—number of vertices, number of edges, and degrees of vertices—all agree for the two graphs. However, H has a simple circuit of length three, namely, v_1, v_2, v_6, v_1, whereas G has no simple circuit of length three, as can be determined by inspection (all simple circuits in G have length at least four). Because the existence of a simple circuit of length three is an isomorphic invariant, G and H are not isomorphic. ◀

THEOREM 2 Let G be a graph with adjacency matrix $\mathbf{A}$ with respect to the ordering $v_1, v_2, \ldots, v_n$ of the vertices of the graph (with directed or undirected edges, with multiple edges and loops allowed). The number of different paths of length r from v_i to v_j, where r is a positive integer, equals the (i, j)th entry of $\mathbf{A}^r$.

Proof: The theorem will be proved using mathematical induction. Let G be a graph with adjacency matrix $\mathbf{A}$ (assuming an ordering $v_1, v_2, \ldots, v_n$ of the vertices of G). The number of paths from v_i to v_j of length 1 is the (i, j)th entry of $\mathbf{A}$, because this entry is the number of edges from v_i to v_j.

Assume that the (i, j)th entry of $\mathbf{A}^r$ is the number of different paths of length r from v_i to v_j. This is the inductive hypothesis. Because $\mathbf{A}^{r+1} = \mathbf{A}^r\mathbf{A}$, the (i, j)th entry of $\mathbf{A}^{r+1}$ equals

$$b_{i1}a_{1j} + b_{i2}a_{2j} + \cdots + b_{in}a_{nj},$$

where b_{ik} is the (i, k)th entry of $\mathbf{A}^r$. By the inductive hypothesis, b_{ik} is the number of paths of length r from v_i to v_k.

A path of length $r + 1$ from v_i to v_j is made up of a path of length r from v_i to some intermediate vertex v_k, and an edge from v_k to v_j. By the product rule for counting, the number of such paths is the product of the number of paths of length r from v_i to v_k, namely, b_{ik}, and the number of edges from v_k to v_j, namely, a_{kj}. When these products are added for all possible intermediate vertices v_k, the desired result follows by the sum rule for counting. ◁

EXAMPLE 15 How many paths of length four are there from a to d in the simple graph G in Figure 8?

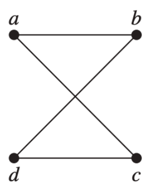


FIGURE 8 **The graph G.**

Solution: The adjacency matrix of G (ordering the vertices as a, b, c, d) is

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$

Hence, the number of paths of length four from a to d is the $(1, 4)$th entry of $\mathbf{A}^4$. Because

$$\mathbf{A}^4 = \begin{bmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{bmatrix},$$

Extra Examples

there are exactly eight paths of length four from a to d. By inspection of the graph, we see that a, b, a, b, d; a, b, a, c, d; a, b, d, b, d; a, b, d, c, d; a, c, a, b, d; a, c, a, c, d; a, c, d, b, d; and a, c, d, c, d are the eight paths of length four from a to d. ◀

Euler and Hamilton Paths

10.5.2 Euler Paths and Circuits

D1 : what is a `Euler circuit`

An Euler circuit in a graph G is **a simple circuit containing every edge of G**. An Euler path in G is **a simple path containing every edge of G**.

EXAMPLE 1 Which of the undirected graphs in Figure 3 have an Euler circuit? Of those that do not, which have an Euler path?

Solution: The graph G_1 has an Euler circuit, for example, a, e, c, d, e, b, a. Neither of the graphs G_2 or G_3 has an Euler circuit (the reader should verify this). However, G_3 has an Euler path, namely, a, c, d, e, b, d, a, b. G_2 does not have an Euler path (as the reader should verify). ◀

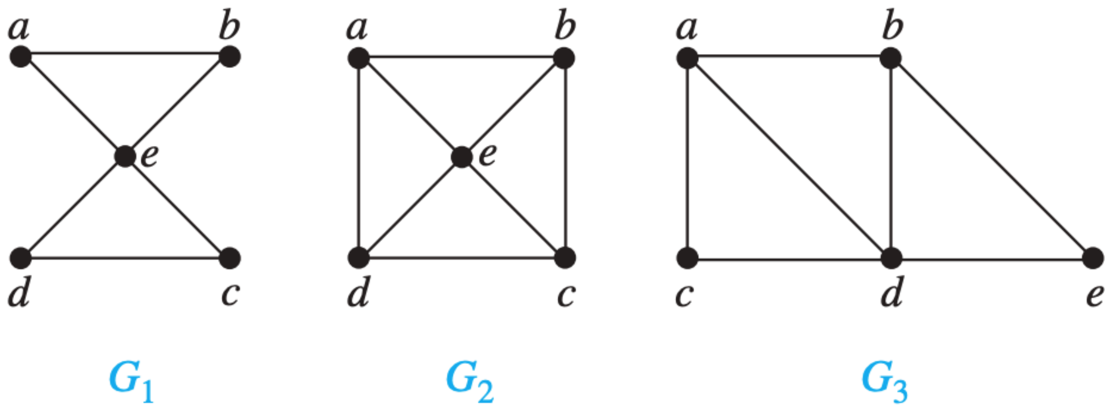


FIGURE 3 **The undirected graphs G_1, G_2, and G_3.**

Theorem 1 :

A connected multigraph with at least two vertices has an Euler circuit if and only if **each of its vertices has even degree.**

ALGORITHM 1 Constructing Euler Circuits.

procedure *Euler*(G: connected multigraph with all vertices of even degree)
circuit := a circuit in G beginning at an arbitrarily chosen vertex with edges successively added to form a path that returns to this vertex
H := G with the edges of this circuit removed
while H has edges
    subcircuit := a circuit in H beginning at a vertex in H that also is an endpoint of an edge of *circuit*
    H := H with edges of *subcircuit* and all isolated vertices removed
    circuit := *circuit* with *subcircuit* inserted at the appropriate vertex
return *circuit* {*circuit* is an Euler circuit}

In a word, the algorithm just visit all subcircuit one by one

the worst case complexity of this algorithm is O(m), where m is the number of edges of G.

> **Theorem 2:**
>
> A connected multigraph has an Euler path but not an Euler circuit if and only if it has **exactly two vertices of odd degree.**

10.5.3 Hamilton Paths and Circuits

> **D2 : `Hamilton Paths`**
>
> A simple path in a graph G that passes through every vertex exactly once is called a Hamilton path, and a simple circuit in a graph G that passes through every vertex exactly once is called a Hamilton circuit. That is, the simple path $x_0, x_1, \ldots, x_{n-1}, x_n$ in the graph $G = (V, E)$ is a Hamilton path if $V = \{x_0, x_1, \ldots, x_{n-1}, x_n\}$ and $x_i \neq x_j$ for $0 \leq i < j \leq n$, and the simple circuit $x_0, x_1, \ldots, x_{n-1}, x_n, x_0$ (with $n > 0$) is a Hamilton circuit if $x_0, x_1, \ldots, x_{n-1}, x_n$ is a Hamilton path.

EXAMPLE 5 Which of the simple graphs in Figure 10 have a Hamilton circuit or, if not, a Hamilton path?

Extra Examples

Solution: G_1 has a Hamilton circuit: a, b, c, d, e, a. There is no Hamilton circuit in G_2 (this can be seen by noting that any circuit containing every vertex must contain the edge $\{a, b\}$ twice), but G_2 does have a Hamilton path, namely, a, b, c, d. G_3 has neither a Hamilton circuit nor a Hamilton path, because any path containing all vertices must contain one of the edges $\{a, b\}$, $\{e, f\}$, and $\{c, d\}$ more than once. ◀

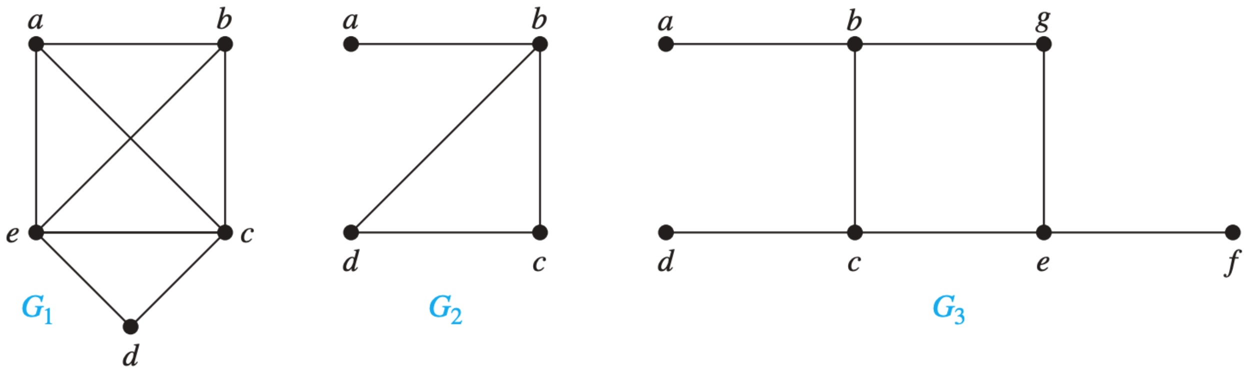


FIGURE 10 Three simple graphs.

THEOREM 3 **DIRAC'S THEOREM** If G is a simple graph with n vertices with $n \geq 3$ such that the degree of every vertex in G is at least $n/2$, then G has a Hamilton circuit.

THEOREM 4 **ORE'S THEOREM** If G is a simple graph with n vertices with $n \geq 3$ such that $\deg(u) + \deg(v) \geq n$ for every pair of nonadjacent vertices u and v in G, then G has a Hamilton circuit.