Unit IV: Graph Theory **07 Hours**

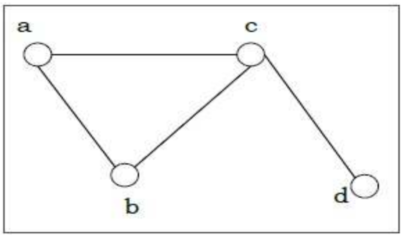
Graphs and Graph Models, Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Single source shortest path-Dikjtra's Algorithm, Planar Graphs, Graph Colouring. Case Study- Web Graph, Google map.

❖ INTRODUCTION

- Graphs are discrete structures consisting of vertices and edges that connect these vertices.
- There are different kinds of graphs, depending on whether edges have directions, whether multiple edges can connect the same pair of vertices, and whether loops are allowed.

❖ Definition

- **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.
- **Example 1:** Let us consider, a Graph is G = (V, E) where V = {a, b, c, d} and E = {{a, b}, {a, c}, {b, c},{c, d}}

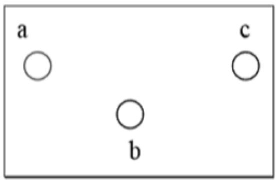
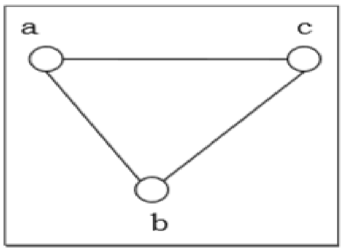
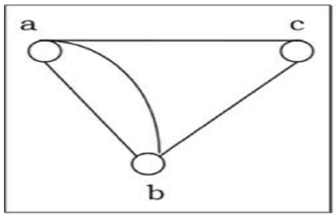
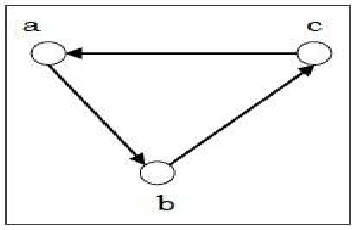
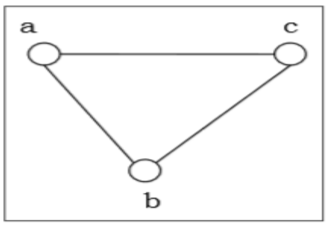
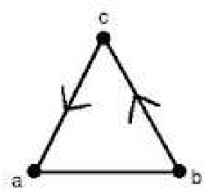
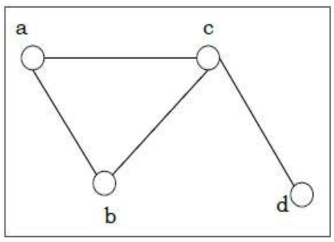


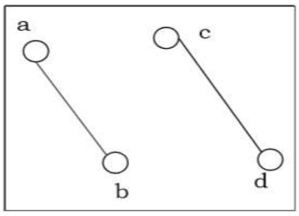
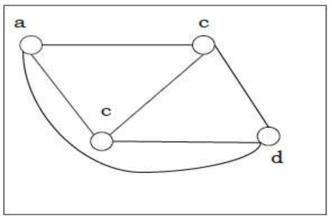
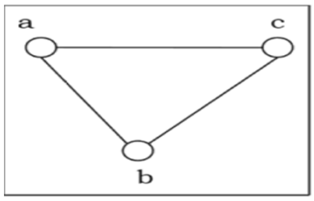
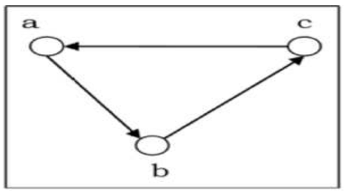
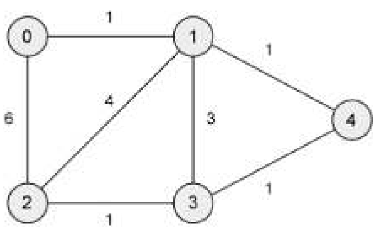
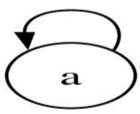
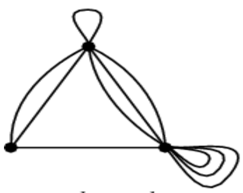
- **Even and Odd Vertex** − If the degree of a vertex is even, the vertex is called an even vertex and if the degree of a vertex is odd, the vertex is called an odd vertex.
- **Degree of a Vertex** − The degree of a vertex V of a graph G (denoted by deg (V)) is the number of edges incident with the vertex V.
- **The Handshaking Lemma**− In a graph, the sum of all the degrees of vertices is equal to twice the number of edges.

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

Vertex	Degree	Even / Odd
a	2	even
b	2	even
c	3	odd
d	1	odd

❖ **Types of Graphs**

Null Graph	A null graph has no edges. The null graph of n vertices is denoted by Nn	
Simple Graph	A graph is called simple graph/strict graph if the graph is undirected and does not contain any loops or multiple edges. In a simple graph each edge connects two different vertices and no two edges connect the same pair of vertices.	
Multi-Graph	If in a graph multiple edges between the same set of vertices are allowed, it is called Multigraph. When m different edges connect the vertices u and v, we say that {u,v} is an edge of multiplicity m.	
Directed Graph	A graph G = (V, E) is called a directed graph if the edge set is made of ordered vertex pair.	
Undirected Graph	A graph G = (V, E) is called a undirected if the edge set is made of unordered vertex pair.	
Mixed Graph	A graph with both directed and undirected edges is called a mixed graph.	
Connected Graph	A graph is connected if any two vertices of the graph are connected by a path	

Disconnected Graph	A graph is disconnected if at least two vertices of the graph are not connected by a path. If a graph G is unconnected, then every maximal connected subgraph of G is called a connected component of the graph G.	
Regular Graph	A graph is regular if all the vertices of the graph have the same degree. In a regular graph G of degree r, the degree of each vertex of G is r.	
Complete Graph	A graph is called complete graph if every two vertices pair are joined by exactly one edge. The complete graph with n vertices is denoted by Kn	
Cycle Graph	If a graph consists of a single cycle, it is called cycle graph. The cycle graph with n vertices is denoted by Cn	
Infinite graph	A graph with an infinite vertex set or an infinite number of edges is called an infinite graph	
Finite graph	A graph with a finite vertex set and a finite edge set is called a finite graph	
Weighted Graph	A graph having a weight, or number, associated with each edge.	
Loop	An edge that connects a vertex to itself is called a loop.	
Pseudograph	A pseudograph may include loops, as well as multiple edges connecting the same pair of vertices.	 *pseudograph*

❖ Graph Terminology

- **Adjacent Vertex**

  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.

- **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.

- **Degree of a Vertex**

  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).

- **Isolated and Pendent**

  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.

- **Initial and Terminal Vertex**

  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.

- **In-degree and Out-degree**

  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.).

- **Theorem 1**: The 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.)

- **Theorem 2:** An undirected graph has an even number of vertices of odd degree.

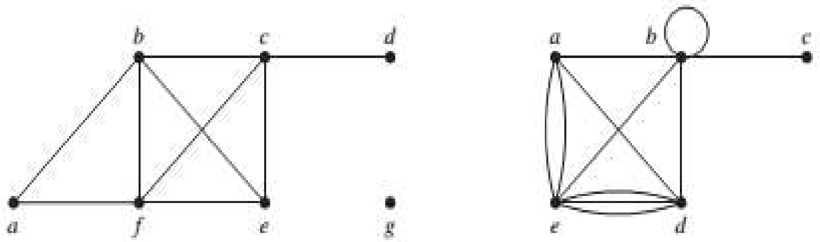
Proof: Let V_1 and V_2 be the set of vertices of even degree and the set of vertices of odd degree, respectively, in an undirected graph G = (V , E) with m edges. Then

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

- **Theorem 3:** Let G = (V, E) be a graph with directed edges. Then

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

- **Example 1:** What are the degrees and what are the neighborhoods of the vertices in the graphs G and H displayed in figure below.

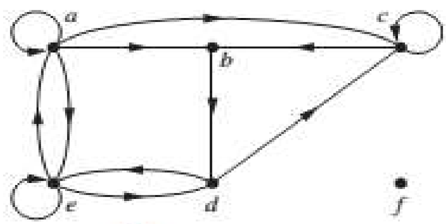


G H

Solution:

In Graph G, deg(a) = 2, deg(b) = deg(c) = deg(f) = 4, deg(d) = 1, deg(e) = 3, and deg(g) = 0.	**In Graph 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, 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)=∅.	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}.

- **Example 2:** Find the in-degree and out-degree of each vertex in the graph G with directed edges shown in figure below.

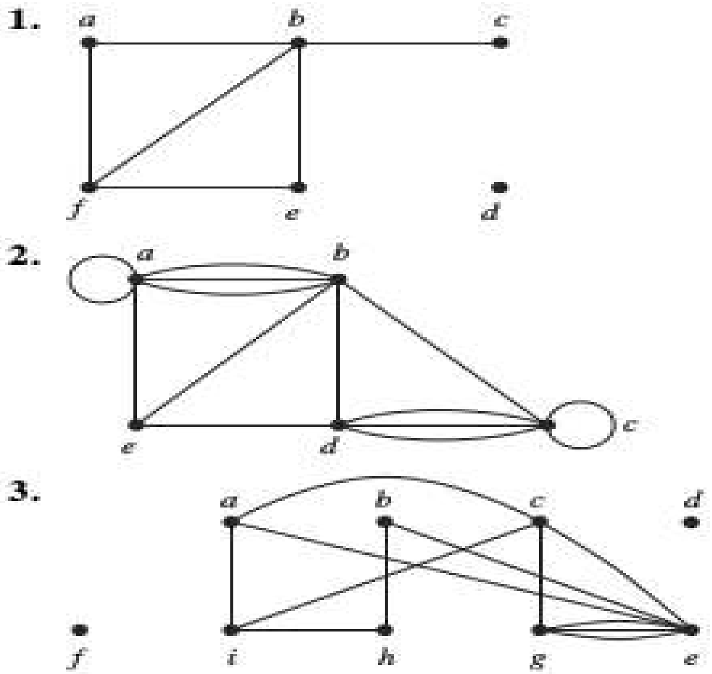


G

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 $deg^{+}(f) = 0$.

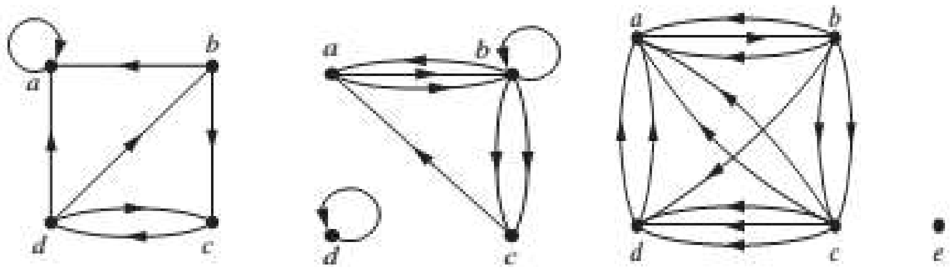
➢ **Example 3:** Find the number of vertices, the number of edges, and the degree of each vertex in the given undirected graph. Identify all isolated and pendant vertices.



Solution:

1. There are 6 vertices here, and 6 edges. The degree of each vertex is the number of edges incident to it. Thus deg(a) = 2, deg(b) = 4, deg(c) = 1 (and hence c is pendant), deg(d) = 0 (and hence d is isolated), deg(e) = 2, and deg(!)= 3. Note that the sum of the degrees is 2 + 4 + 1+0 + 2 + 3 = 12, which is twice the number of edges.
2. There are 5 vertices and 13 edges. Thus deg(a) = 6, deg(b) = 6, deg(c) = 6, deg(d) = 5 and deg(e) = 3. The sum of the degrees is 5 + 6 + 5 + 5 + 3= 26.
3. There are 9 vertices here, and 12 edges. The degree of each vertex is the number of edges incident to it. Thus deg(a) = 3, deg(b) = 2, deg(c) = 4, deg(d) = 0 (and hence dis isolated), deg(e) = 6, deg(f) = 0 (and hence f is isolated), deg(g) = 4, deg(h) = 2, and deg(i) = 3. Note that the sum of the degrees is 3 + 2 + 4 + 0 + 6 + 0 + 4 + 2 + 3 = 24, which is twice the number of edges.

➢ **Example 4:** Determine the number of vertices and edges and find the in-degree and out-degree of each vertex for the given directed multi-graph. Also determine sum of the in-degrees of the vertices and the sum of the out-degrees of the vertices directly.



Solution:

1. This directed graph has 4 vertices and 7 edges.

   The in-degree of vertex a is $deg^{-}(a) = 3$ and out-degree is $deg^{+}(a) = 1$

   Similarly for $deg^{-}(b) = 1$, $deg^{+}(b) = 2$, $deg^{-}(c) = 2$, $deg^{+}(c) = 1$, $deg^{-}(d) = 1$, and $deg^{+}(d) = 3$.

   The sum of the in-degrees and the sum of the out-degrees are equal i.e. 7

2. This directed graph has 4 vertices and 8 edges.

   The in-degree of vertex a is $deg^{-}(a) = 2$ and out-degree is $deg^{+}(a) = 2$.

   Similarly we have $deg^{-}(b) = 3$, $deg^{+}(b) = 4$, $deg^{-}(c) = 2$, $deg^{+}(c) = 1$, $deg^{-}(d) = 1$, $deg^{+}(d) = 1$,

   The sum of the in-degrees and the sum of the out-degrees are both equal to the number of edges (8)

3. This directed multigraph has 5 vertices and 13 edges.

   The in-degree of vertex a is $deg^{-}(a) = 6$ and out-degree is $deg^{+}(a) = 1$.

   Similarly we have deg-(b) = 1, $deg^{+}(b) = 5$, $deg^{-}(c) = 2$, $deg^{+}(c) = 5$, $deg^{-}(d) = 4$, $deg^{+}(d) = 2$, $deg^{-(}e) = 0$, and $deg^{+}(e) = 0$ (vertex e is isolated).

   The sum of the in-degrees and the sum of the out-degrees are both equal to the number of edges (13).

- **Example 5:** 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 * 10 = 60, it follows that 2m = 60 where m is the number of edges. Therefore, m=30.

- **Example 6:** How many vertices does a regular graph of degree four with 10 edges have?
  Solution: If a graph is regular of degree 4 and has n vertices, then by the handshaking theorem it has 4n/2 = 2n edges. Since we are told that there are 10 edges, we just need to solve 2n = 10. Thus the graph has 5 vertices. The complete graph K_5 is one such graph.

❖ Representation of Graphs

A. Adjacency List

B. Adjacency Matrix

C. Incidence Matrix

A. Adjacency List

- 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 7:** Use adjacency lists to describe the simple graph given in Figure 1.

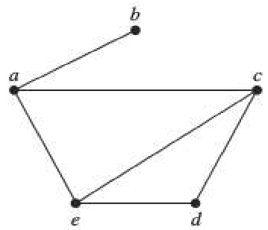


FIGURE 1 **A Simple Graph.**

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 8:** 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.

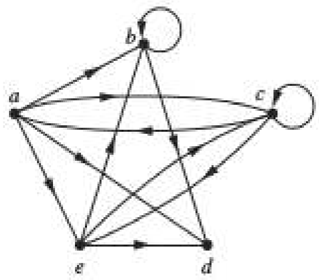


FIGURE 2 **A Directed Graph.**

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*

- **Drawback of Adjacency List:** Carrying out graph algorithms using the representation of graphs by lists of edges, or by adjacency lists, can be bulky if there are many edges in the graph.

B. Adjacency Matrices

- Suppose that G = (V, E) is a simple graph where |V | = n. Suppose that the vertices of G are listed arbitrarily as v1 , v2 , . . . , vn . 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 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 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 9:** 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

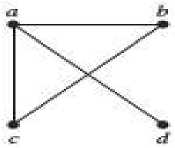


FIGURE 3
Simple Graph.

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

- **Example 10:** Draw a graph with the adjacency matrix with respect to the ordering of vertices a, b, c, d.

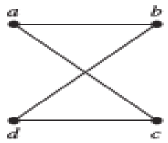
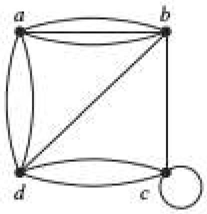


FIGURE 4
A Graph with the Given Adjacency Matrix.

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

- **Example 11:** Use an adjacency matrix to represent the pseudograph shown in Figure.



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

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

- **Important Note:**
  - ✓ 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, **aij = aji** , because both of these entries are 1 when vi and vj are adjacent, and both are 0 otherwise. Furthermore, because a simple graph has no loops, each entry aii , i = 1, 2, 3, . . . , n, is 0.
  - ✓ Adjacency matrices can also be used to represent undirected graphs with loops and with multiple edges. A loop at the vertex vi is represented by a 1 at the (i, i)th position of the adjacency matrix. When multiple edges connecting the same pair of vertices vi and vj , 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 {vi , vj}. All undirected graphs, including multigraphs and pseudographs, have symmetric adjacency matrices.

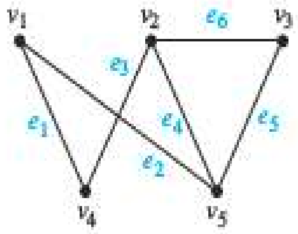
C. Incidence Matrices

- 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 = [mij], where

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

- 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 12:** Represent the graph shown in below with an incidence matrix.

Solution: The incidence matrix is



	e_1	e_2	e_3	e_4	e_5	e_6
v_1	1	1	0	0	0	0
v_2	0	0	1	1	0	1
v_3	0	0	0	0	1	1
v_4	1	0	1	0	0	0
v_5	0	1	0	1	1	0

- **Example 13:** Represent the pseudograph shown below using an incidence matrix.

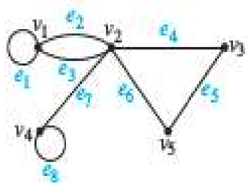


FIGURE 7
A Pseudograph.

Solution: The incidence matrix for this graph is

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
v_1	1	1	1	0	0	0	0	0
v_2	0	1	1	1	0	1	1	0
v_3	0	0	0	1	1	0	0	0
v_4	0	0	0	0	0	0	1	1
v_5	0	0	0	0	1	1	0	0

- **Example 14:** Represent each of these graphs with an adjacency matrix.

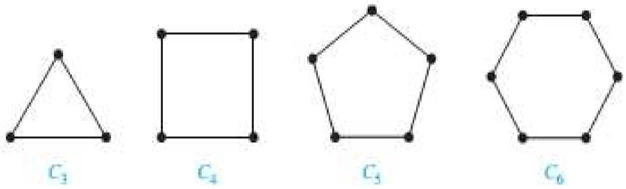
  a) K_4 b) $K_{1,4}$ c) $K_{2,3}$

  d) C_4 e) W_4 f) Q_3

Solution:

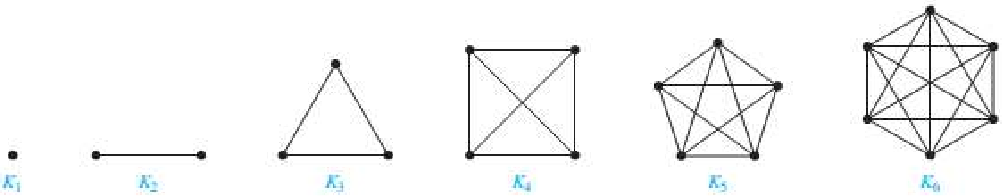
❖ Some Special Simple Graphs

- **Cycles:** A cycle C_n, $n \geq 3$, consists of n vertices $v_1, v_2, \ldots, vn$ and edges $\{v_1, v_2\}$, $\{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}$, and $\{vn, v1\}$. The cycles C_3, C_4, C_5, and C_6 is shown below:

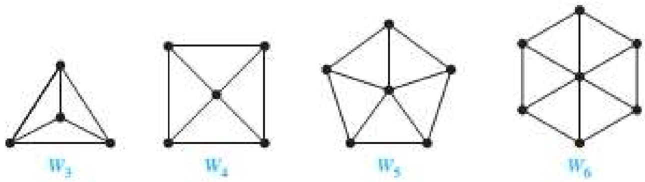


- **Complete Graphs:** A complete graph on n vertices, denoted by K_n, is a simple graph that contains exactly one edge between each pair of distinct vertices.

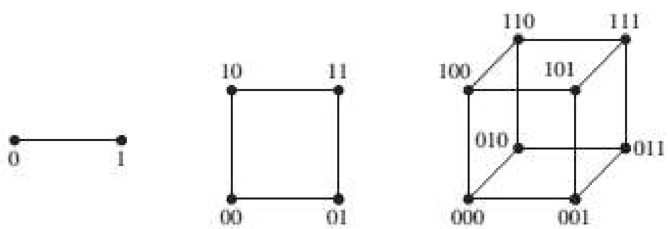
  A simple graph for which there is at least one pair of distinct vertex not connected by an edge is called noncomplete.



- **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 then vertices in C_n, by new edges. The wheels W_3, W_4, W_5, and W_6 are displayed in Figure below.



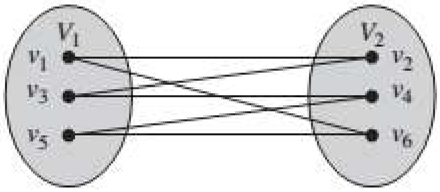
- **N-Cubes:** An n-dimensional hypercube, or n-cube, denoted by Q_n, is a graph that has vertices representing the 2^n bit strings of length *n*. Two vertices are adjacent if and only if the bit strings that they represent differ in exactly one bit position. Q_1, Q_2, and Q_3 in Figure.



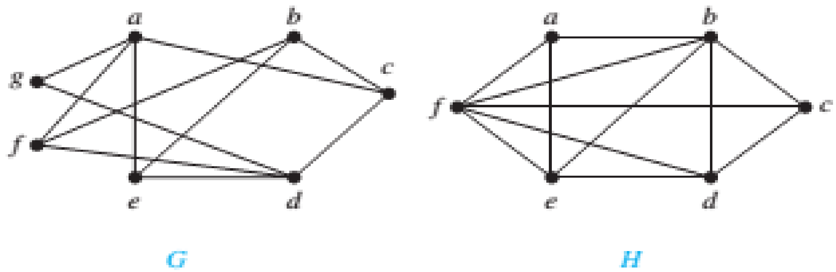
❖ Bipartite Graphs

- A simple graph G is called bipartite if its vertex set V can be partitioned into two disjoint sets V_1 andV_2 such that every edge in the graph connects a vertex in V_1 and a vertex in V_2 (so that no edge in G connects either two vertices in V_1 or two vertices in

V_2). When this condition holds, we call the pair (V_1, V_2) a bipartition of the vertex set V of G.



- C_6 is bipartite, as shown in Figure above, because its vertex set can be partitioned into the two sets V_1= {v1, v3, v5} and V_2= {v2, v4, v6}, and every edge of C_6 connects a vertex in V_1 and a vertex in V_2.
- K_3 is not bipartite. To verify this, note that if we divide the vertex set of K_3 into two disjoint sets, one of the two sets must contain two vertices. If the graph were bipartite, these two vertices could not be connected by an edge, but in K_3 each vertex is connected to every other vertex by an edge.
- **Example 15:** Are the graphs G and H displayed in Figure below bipartite?



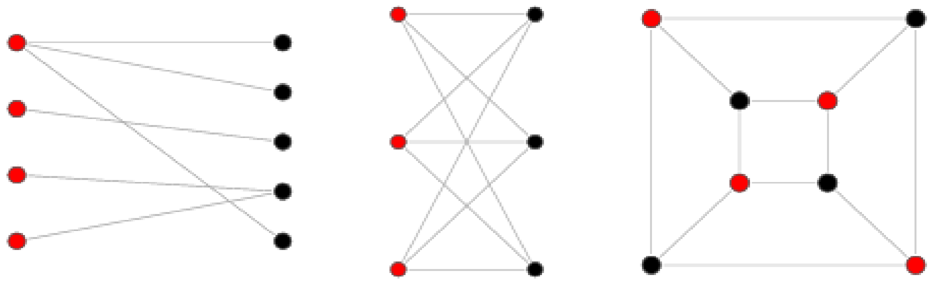
Solution: Graph G is bipartite because its vertex set is the union of two disjoint sets,{a, b, d} and{c, e, f, g}, and each edge connects a vertex in one of these subsets to a vertex in the other subset. (Note: For G to be bipartite it is not necessary that every vertex in {a, b, d}be adjacent to every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

- **Theorem 4:** 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.

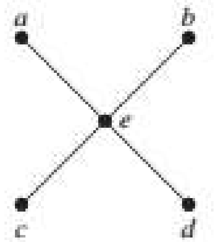
Proof:

- ✓ First, suppose that G = (V, E) is a bipartite simple graph. Then $V = V_1 \cup V_2$, where V_1 and V_2 are disjoint sets and every edge in E connects a vertex in V_1 and a vertex in V_2.
- ✓ If we assign one color to each vertex in V_1 and a second color to each vertex in V_2, then no two adjacent vertices are assigned the same color.

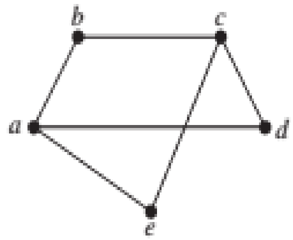
- **Algorithm to find out whether a given graph is Bipartite**
  1. Assign RED color to the source vertex (putting into set V_1).
  2. Color all the neighbors with BLUE color (putting into set V_2).
  3. Color all neighbor's neighbor with RED color (putting into set V_1).
  4. This way, assign color to all vertices such that it satisfies all the constraints of m way coloring problem where m = 2.
  5. While assigning colors, if we find a neighbor which is colored with same color as current vertex, then the graph cannot be colored with 2 vertices (or graph is not Bipartite).
- **Example 16:** Are the following graphs bipartite. If Yes Justify answer.



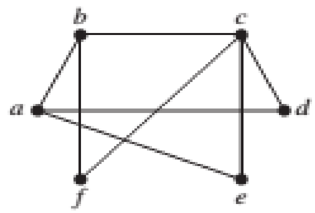
- **Example 17:** Determine whether the graph is bipartite.



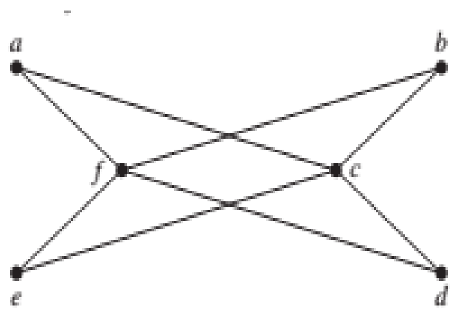
(a)



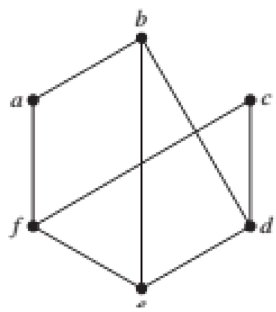
(b)



(c)



(d)

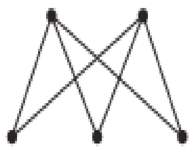


(e)

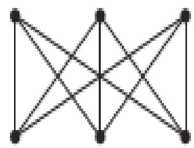
Solution: Bipartite: a), b) & d)

❖ Complete Bipartite Graphs

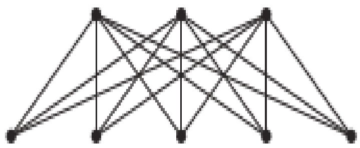
- A complete bipartite graph $K_{m,n}$ is a graph that has its vertex set partitioned into two subsets of *m* and *n* vertices, respectively with an edge between two vertices if and only if one vertex is in the first subset and the other vertex is in the second subset.
- i.e. each vertex of Vi is joined to every vertex of Vj by an unique edge.



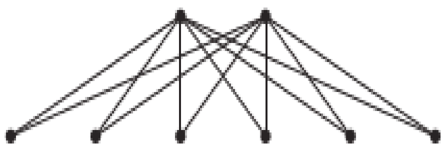
$K_{2,3}$



$K_{3,3}$



$K_{3,5}$



$K_{2,6}$

➢ **Example 18:** For which values of n are these graphs bipartite?

a) K_n b) C_n c) W_n d) Q_n

Solution:

a) K_n is bipartite if and only if $n = 2$

b) C_n is bipartite if and only if n is even.

c) W_n is not bipartite for any n.

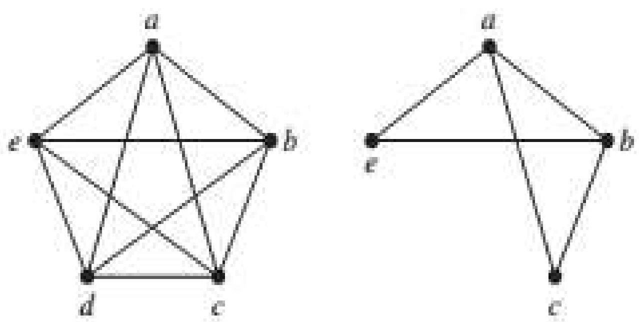
d) Q_n is bipartite for all $n \geq 2$.

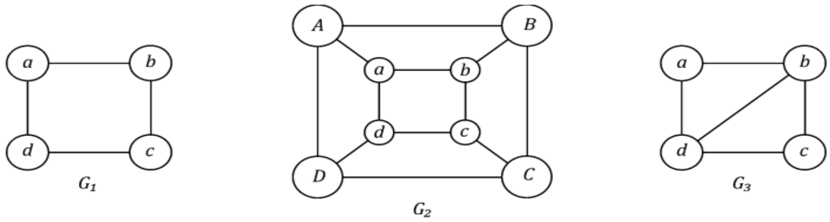
❖ New Graphs from Old

➢ A **subgraph of a graph** G = (V, E) is a graph H = (W, F), where $W \subseteq V$ and $F \subseteq E$. A subgraph H of G is a proper subgraph of G if $H \neq G$.
Given a set of vertices of a graph, we can form a subgraph of this graph with these vertices and the edges of the graph that connect them.

➢ 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.

➢ **Example 19:** The graph G shown in Figure is a subgraph of K_5. If we add the edge connecting *c* and *e* to G, we obtain the subgraph induced by W= {a, b, c, e}.





$G_1 \subseteq G_2$, $G_1 \subseteq G_3$ but $G_3 \nsubseteq G_2$.

- **Removing or Adding Edges of a Graph**

  Given a graph G = (V, E) and an edge e ∈ E, we can produce a subgraph of G by removing the edge e. The resulting subgraph, denoted by G − e, has the same vertex set V as G. Its edge set is E − e. Hence, **G − e = (V, E − {e})**

  Add an edge *e* to a graph to produce a new larger graph when this edge connects two vertices already in G. We denote by G + e the new graph produced by adding a new edge e, connecting two previously non incident vertices, to the graph G. Hence,

  G + e = (V, E ∪ {e})

  The vertex set of G + e is the same as the vertex set of G and the edge set is the union of the edge set of G and the set {e}.

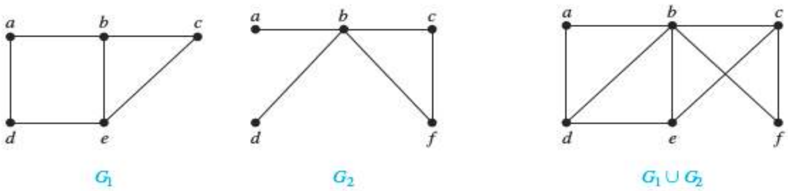
- **Removing Vertices from a Graph**

  When we remove a vertex v and all edges incident to it from G = (V, E), we produce a subgraph, denoted by G − v. Observe that **G − v = (V − v, E')**, where E' is the set of edges of G not incident to v.

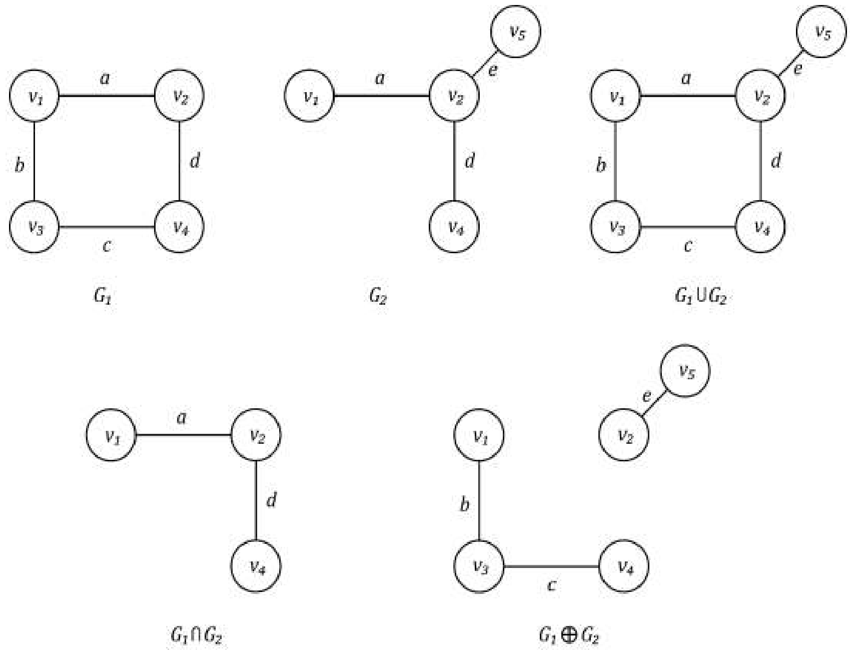
  Similarly, if V' is a subset of V, then the graph G – V' is the subgraph (V – V', E), where E' is the set of edges of G not incident to a vertex in V'.

❖ **Operations on Graphs**

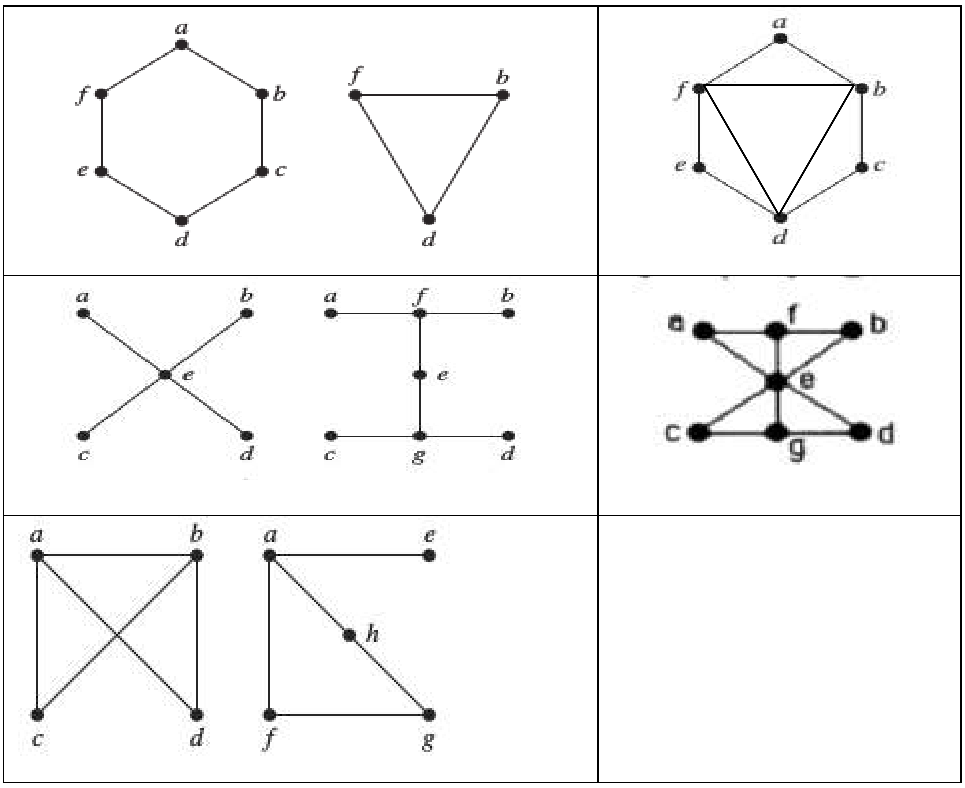
- **Union:** The union of two simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the simple graph with vertex set $V_3 = V_1 \cup V_2$ and edge set $E_3 = E_1 \cup E_2$. The union of G_1 and G_2 is denoted by $G_3 = G_1 \cup G_2$.
- **Intersection:** The intersection of two graphs G_1 and G_2 denoted by $G_1 \cap G_2$ is a graph G4 consisting only of those vertices and edges that are in both G_1 and G_2.
- **Ring:** The ring sum of two graphs G_1 and G_2, denoted by $G_1 \oplus G_2$; is a graph consisting of the vertex set $V_1 \cup V_2$ and of edges that are either in G_1 or G_2; but not in both.
- **Example 20:** Find the union of the graphs G_1 and G_2 shown in Figure.



- **Example 21:**



- **Example 22:** Find the union of the graphs shown in Figure below.



❖ 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?
- 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.

❖ **Determining whether two Simple Graphs are Isomorphic**

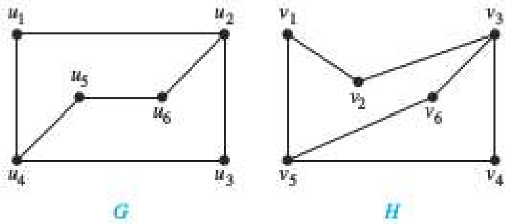
- It is often difficult to determine whether two simple graphs are isomorphic.
- There are n! possible one-to-one correspondences between the vertex sets of two simple graphs with n vertices.
- Testing each such correspondence to see whether it preserves adjacency and non-adjacency is impractical if n is at all large.
- 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**.
- 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.
- 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.
- Isomorphic simple graphs also must have the **same degrees of the vertices**. 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.

Note: If two graphs are isomorphic, they must have:

  - Must have same number of vertices
  - Must have same number of edges
  - Must have equal number of vertices with same degree.
  - Must have equal number of loops
  - Must have equal number of pendent
  - G_1 and G_2 must have equal number of pendent edges.
  - If u and v are adjacent in G_1 then the corresponding vertices in G_2 are also adjacent.

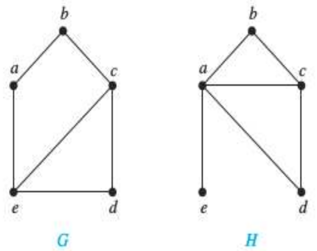
- In general, it is easier to prove two graphs are not isomorphic by proving that one of the above properties fails.

- **Example 23:** Determine whether the graphs G and H displayed are isomorphic.

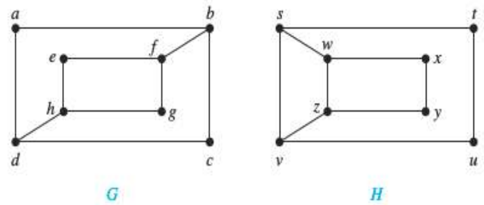


Solution: Both G and H have six vertices and seven edges. Both have four vertices of degree two and two vertices of degree three. It is also easy to see that the subgraphs of G and H consisting of all vertices of degree two and the edges connecting them are isomorphic.

- **Example 24:** Determine whether the graphs shown in figure a and b are isomorphic.



(a)



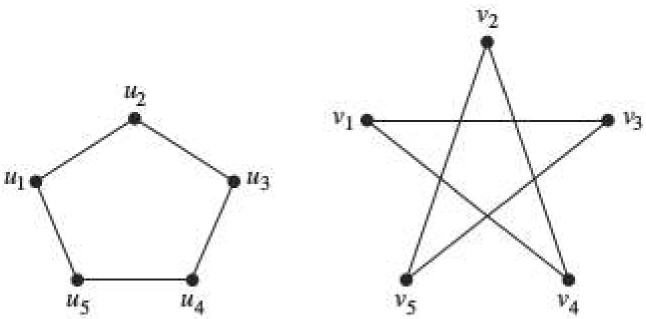
(b)

a) 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 shows that G and H are not isomorphic.

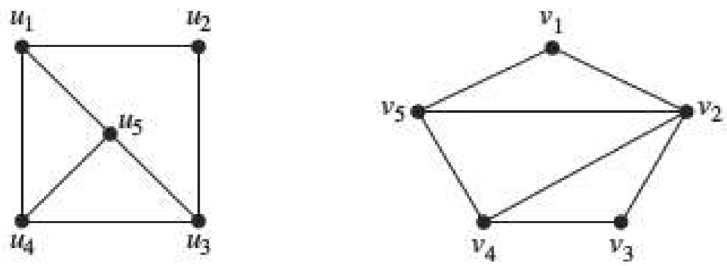
b) The graphs G and H both have eight vertices and 10 edges. They also both have four vertices of degree two and four of degree three. Because these invariants all agree, it is still conceivable that these graphs are isomorphic.

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.

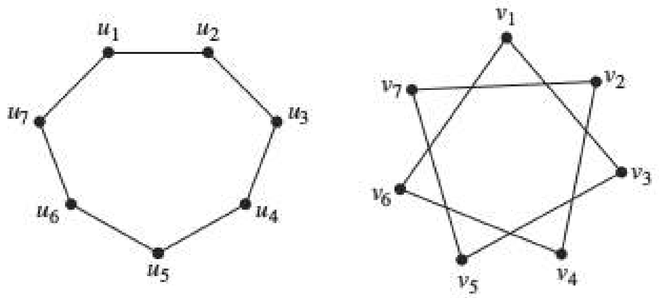
- **Example 25:** Determine whether the following graphs are isomorphic.



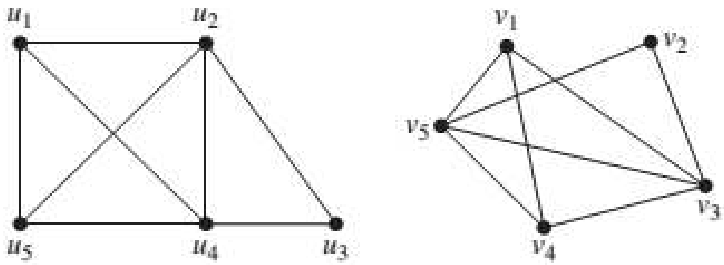
Isomorphic



Non- isomorphic



Isomorphic



- **Example 26:** Are the simple graphs with the following adjacency matrices isomorphic?

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

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

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

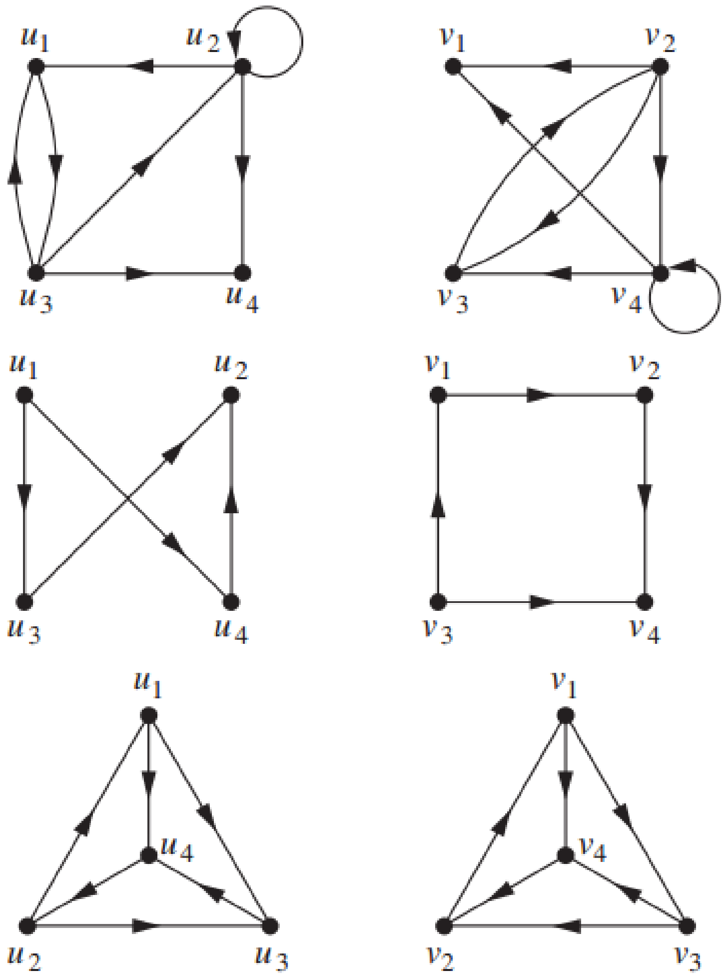
Solution:

a. Both graphs consist of 2 sides of a triangle; they are clearly isomorphic.

b. The graphs are not isomorphic, since the first has 4 edges and the second has 5 edges.

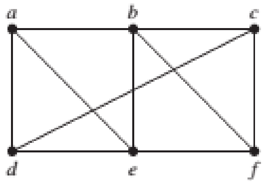
c. The graphs are not isomorphic, since the first has 4 edges and the second has 3 edges.

- **Example 27:** Define isomorphism of directed graphs.



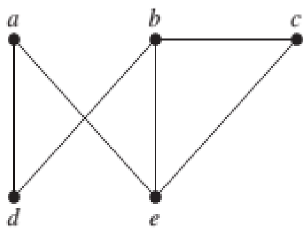
❖ **Connectivity - Paths**

- 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.
- 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.
- A **path** or **circuit** is simple if it does not contain the same edge more than once.
- A **circuit** in a graph is also called as cycle in a graph.
- A **walk** is an alternating sequence of vertices and edges of a graph.
- A **path** is a walk that does not include any vertex twice, except that its first vertex might be the same as its last.
- A **trail** is a walk that does not pass over the same edge twice. A trail might visit the same vertex twice, but only if it comes and goes from a different edge each time.
- **Example 28:** In the simple graph shown below *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.



- **Example 29:** Does each of these lists of vertices form a path in the following graph? Which paths are simple? Which are circuits? What are the lengths of those that are paths?

  i) a, e, b, c, b ii) a, e, a, d, b, c, a iii) e, b, a, d, b, e iv) c, b, d, a, e, c



Solution: i) This is a path of length 4, but it is not simple, since edge {b, c} is used twice. It is not a circuit, since it ends at a different vertex from the one at which it began.

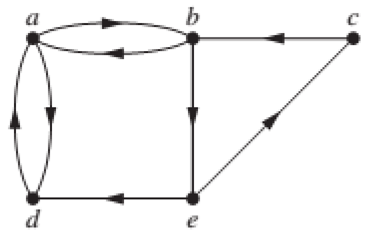
ii) This is not a path, since there is no edge from c to a.

iii) This is not a path, since there is no edge from b to a.

iv) This is a path of length 5 (it has 5 edges in it). It is simple, since no edge is repeated. It is a circuit since it ends at the same vertex at which it began.

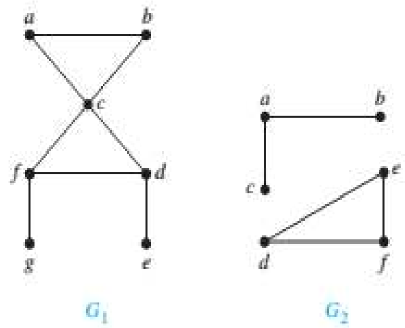
- **Example 30:** Does each of these lists of vertices form a path in the following graph? Which paths are simple? Which are circuits? What are the lengths of those that are paths?

  i) a, b, e, c, b ii) a, d, a, d, a iii) a, d, b, e, a iv) a, b, e, c, b, d, a



❖ Connectedness in Undirected Graphs

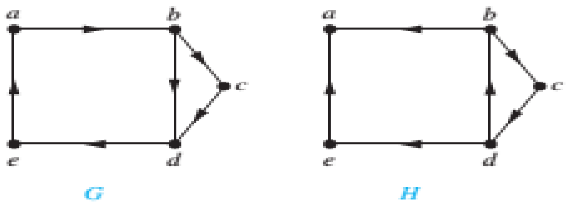
- An undirected graph is called **connected** if there is a path between every pair of distinct vertices of the graph.
- An undirected graph that is not connected is called **disconnected.** We disconnect a graph when we remove vertices or edges, or both, to produce a disconnected subgraph.



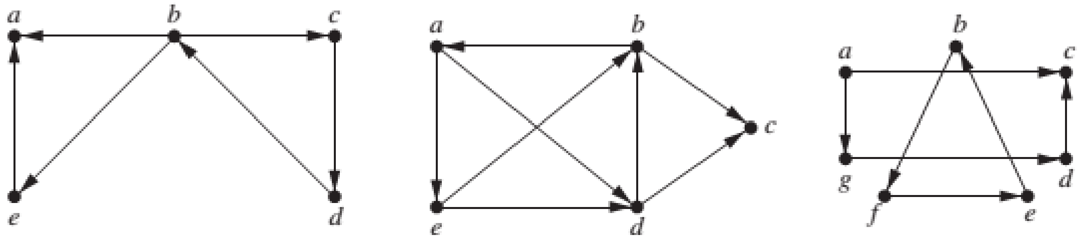
- The graph G_1 in Figure above is connected, because for every pair of distinct vertices there is a path between them.However, the graph G_2 is not connected.

❖ Connectedness in Directed Graphs

- 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.
- 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.



- Graph G is strongly connected because there is a path between any two vertices in this directed graph.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.
- **Example 31:** Determine whether each of these graphs is strongly connected and if not, whether it is weakly connected.



(a) **(b)** **(c)**

Solution: a) Notice that there is no path from a to any other vertex, because both edges involving a are directed toward a. Therefore the graph is not strongly connected. However, the underlying undirected graph is clearly connected, so this graph is weakly connected.

b) Notice that there is no path from c to any other vertex, because both edges involving c are directed toward c. Therefore the graph is not strongly connected. However, the underlying undirected graph is clearly connected, so this graph is weakly connected.

c) The underlying undirected graph is clearly not connected (one component has vertices b, f, and e), so this graph is neither strongly nor weakly connected.

❖ Euler Paths and Circuits

- An **Euler path** is a path that uses every edge of a graph exactly once. An Euler path starts and ends at **different** vertices.
- If a graph G has an Euler path, then it must have exactly two odd vertices.

OR

- If the number of odd vertices in G is anything other than 2, then G cannot have an Euler path.

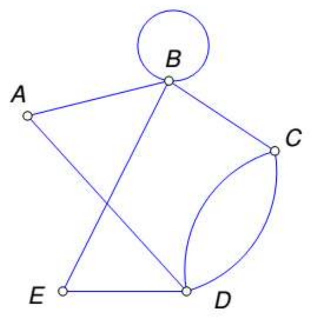
- An **Euler circuit** is a circuit that uses every edge of a graph exactly once. An Euler circuit starts and ends at the **same** vertex.
- If a graph G has an Euler circuit, then all of its vertices must be even vertices.

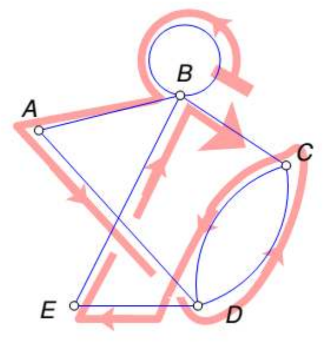
  OR
- If the number of odd vertices in G is anything other than 0, then G cannot have an Euler circuit.
- In **Euler paths and Euler circuits**, the game was to find paths or circuits that include **every edge** of the graph once (and only once).

# Odd Vertices	Euler Path?	Euler Circuit?
0	No	Yes*
2	Yes*	No
4, 6, 8,….	No	No
1, 3, 5,…..	No such graphs exist	

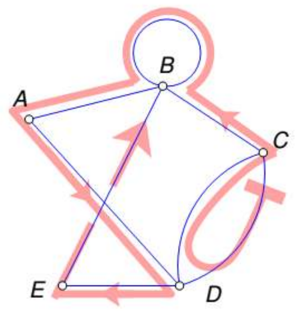
(* Provided the graph is connected)

- **Bridges -** Removing a single edge from a connected graph can make it disconnected. Such an edge is called a bridge.
- Loops cannot be bridges, because removing a loop from a graph cannot make it disconnected.
- If two or more edges share both endpoints, then removing any one of them cannot make the graph disconnected. Therefore, none of those edges is a bridge.
- **Example 32:** Find Euler path and circuit for given graph.

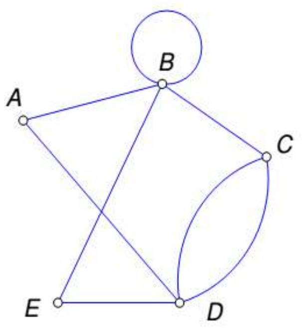


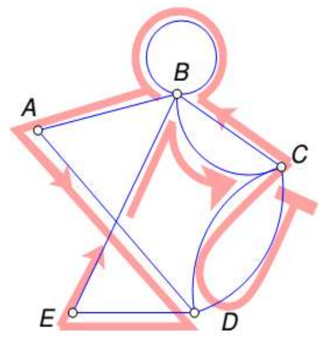


Eular Path : BBADCDEBC

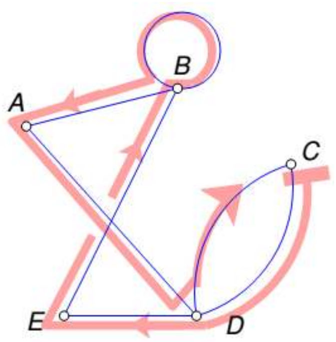


Eular Path : CDCBBADEB





Euler Circuit: CDCBBADEBC



Euler Circuit: CDEBBADC

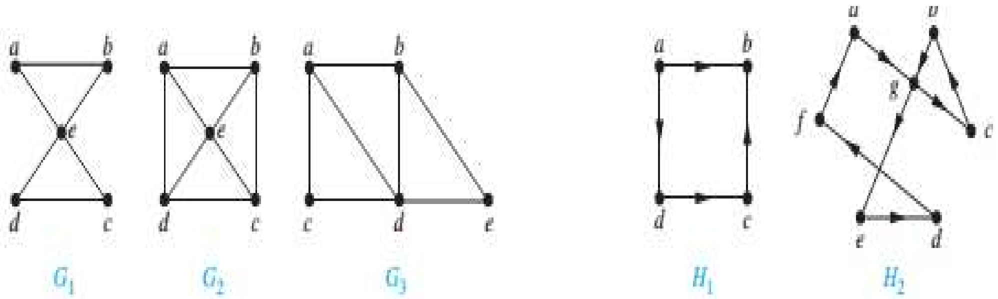
- **Euler path & Euler circuit for directed graphs**

  For a directed graph to have Euler path, on each node, number of incoming edges should be equal to number of outgoing nodes except start node where out degree is one more than in degree and end node where incoming is one more than outgoing.

  To have Euler circuit, all nodes should have in degree equal to out degree.

  We have to keep in mind that for both directed and undirected graphs, above conditions hold when all nodes with non-zero degree are part of strongly connected component of graph.

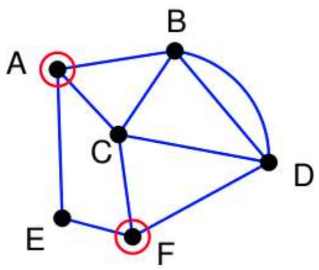
- **Example 33:** Which of the undirected and directed graphs in Figure shown have an Euler circuit? Of those that do not, which have an Euler path.



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. G_1 and G_2 does not have an Euler path. G_3 has an Euler path, namely,a, c, d, e, b, d, a, b.

The graph H_2 has an Euler circuit, for example,a, g, c, b, g, e, d, f, a. Neither H_1 nor H_3 has an Euler circuit. H_3 has an Euler path, namely, c, a, b, c, d, b,but H_1 does not.

- **Example 34:** Finding Euler Circuits and Paths



Solution: Euler Path: FEACBDCFDBA

❖ Hamilton Paths and Circuits

- A **Hamilton path** in a graph is a path that includes each vertex of the graph once and only once.
- A **Hamilton circuit** is a circuit that includes each vertex of the graph once and only once.
- In **Hamilton paths and Hamilton circuits**, the game is to find paths and circuits that include **every vertex** of the graph once and only once.

❖ Hamilton versus Euler

From Figure (a)

- If a graph has a Hamilton circuit, then it automatically has a Hamilton path-(the Hamilton circuit can always be truncated into a Hamilton path by dropping the last vertex of the circuit.)
- Contrast this with the mutually exclusive relationship between Euler circuits and paths: If a graph has an Euler circuit it cannot have an Euler path and vice versa.
- Hamilton circuit is A, F, B, C, G, D, E, A &
- Hamilton path is A, F, B, C, G, D, E.

From Fig (b)

- Has no Euler circuits but does have Euler paths (for example C, D, E, B, A, D) &
- Has no Hamilton circuits (sooner or later you have to go to C, and then you are stuck) but does have Hamilton paths (for example, A, B, E, D, C).
- This illustrates that a graph can have a Hamilton path but no Hamilton circuit!.

From Fig (c)

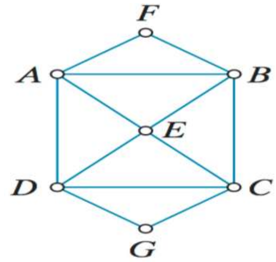
- Has neither Euler circuits nor paths (it has four odd vertices)
- Has Hamilton circuits (for example A, B, C, D, E, A – there are plenty more) and consequently has Hamilton paths (for example, A, B, C, D, E).

From Fig (d)

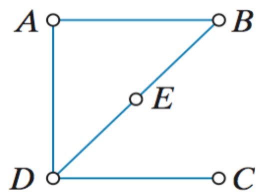
- Has no Euler circuits but has Euler paths (F and G are the two odd vertices) and
- Has neither Hamilton circuits nor Hamilton paths.

From Fig (e)

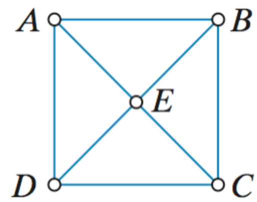
- Has neither Euler circuits nor Euler paths (too many odd vertices) and
- Has neither Hamilton circuits nor Hamilton paths.



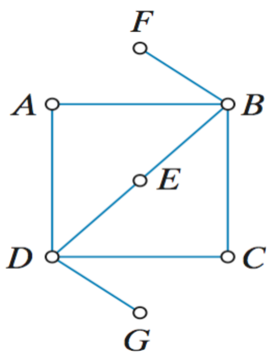
(a)



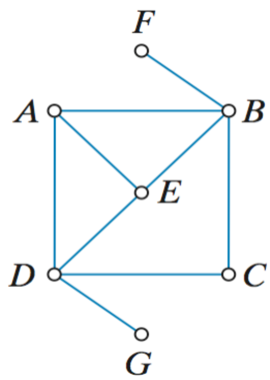
(b)



(c)

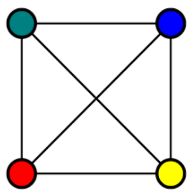


(d)



(e)

- **How Many Hamilton Circuits when a graph have ?**
  - ✓ Consider Complete graph K_n. One of the key properties of K_n is that every vertex has degree n – 1.
  - ✓ This implies that the sum of the degrees of all the vertices is n(n– 1), and it follows from Euler's sum of degrees theorem that the number of edges in K_n is n(n– 1)/2.
  - ✓ For a graph with n vertices and no multiple edges or loops, n(n– 1)/2 is the maximum number of edges possible, and this maximum can only occur when the graph is K_n.
  - ✓ Because K_n has a complete set of edges (every vertex is connected to every other vertex), it also has a complete set of Hamilton circuits –you can travel the vertices in any sequence you choose and you will not get stuck. !
  - ✓ Number of distinct hamilton circuits for K_n graph is **(n-1)!.**
- **Example 35:** Find Hamilton Circuits in K_4.



The Six Hamilton Circuits in K_4

	Reference point is *A*	Reference point is *B*	Reference point is *C*	Reference point is *D*
1	A, B, C, D, A	B, C, D, A, B	C, D, A, B, C	D, A, B, C, D
2	A, B, D, C, A	B, D, C, A, B	C, A, B, D, C	D, C, A, B, D
3	A, C, B, D, A	B, D, A, C, B	C, B, D, A, C	D, A, C, B, D
4	A, C, D, B, A	B, A, C, D, B	C, D, B, A, C	D, B, A, C, D
5	A, D, B, C, A	B, C, A, D, B	C, A, D, B, C	D, B, C, A, D
6	A, D, C, B, A	B, A, D, C, B	C, B, A, D, C	D, C, B, A, D

- **Example 36:** Find Hamilton Circuits in K_5

  Solution: For simplicity, we will write each circuit just once, using a common reference point – say A. (As long as we are consistent, it doesn't really matter which reference point we pick.)

  Each of the Hamilton circuits will be described by a sequence that starts and ends with A, with the letters B, C, D, and E sandwiched in between in some order.

There are 4 ×3 ×2 ×1 = 24 different ways to shuffle the letters B, C, D, and E, each producing a different Hamilton circuit.

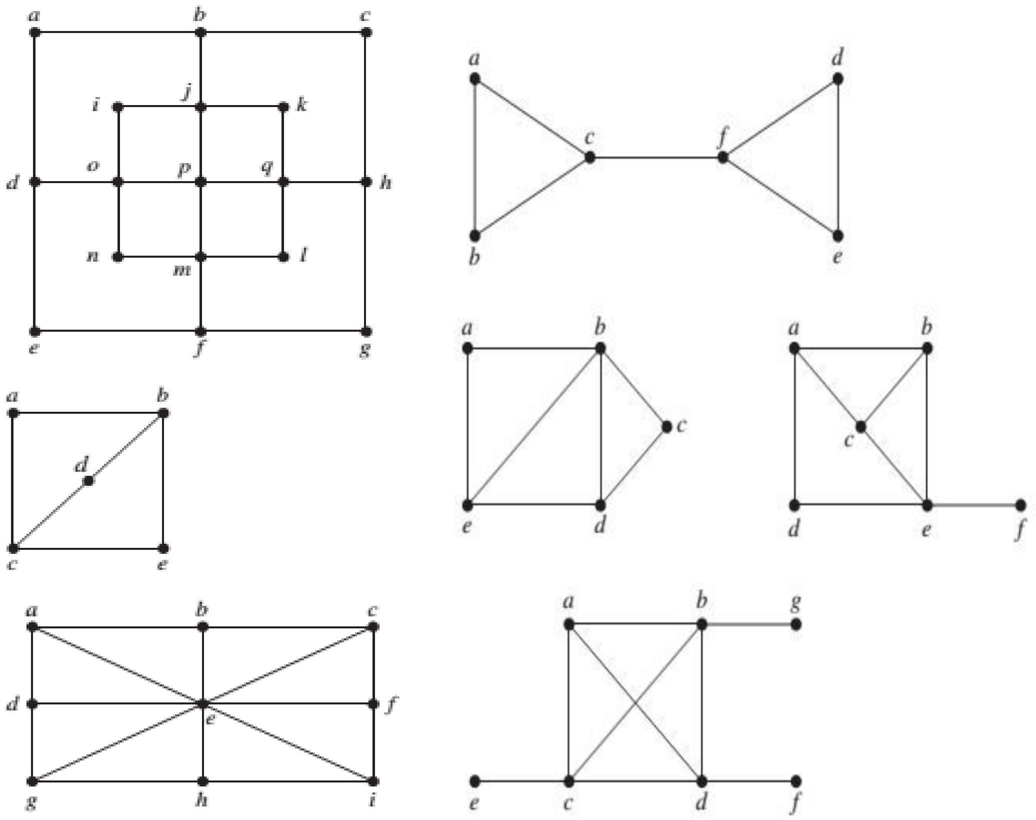
The 24 Hamilton Circuits in K_5

1	A, B, C, D, E, A	13	A, E, D, C, B, A
2	A, B, C, E, D, A	14	A, D, E, C, B, A
3	A, B, D, C, E, A	15	A, E, C, D, B, A
4	A, B, D, E, C, A	16	A, C, E, D, B, A
5	A, B, E, C, D, A	17	A, D, C, E, B, A
6	A, B, E, D, C, A	18	A, C, D, E, B, A
7	A, C, B, D, E, A	19	A, E, D, B, C, A
8	A, C, B, E, D, A	20	A, D, E, B, C, A
9	A, C, D, B, E, A	21	A, E, B, D, C, A
10	A, C, E, B, D, A	22	A, D, B, E, C, A
11	A, D, B, C, E, A	23	A, E, C, B, D, A
12	A, D, C, B, E, A	24	A, E, B, C, D, A

Number of Distinct Hamilton Circuits in K_N

N	$(N-1)!$	N	$(N-1)!$
3	2	12	39,916,800
4	6	13	479,001,600
5	24	14	6,227,020,800
6	120	15	87,178,291,200
7	720	16	1,307,674,368,000
8	5040	17	20,922,789,888,000
9	40,320	18	355,687,428,096,000
10	362,880	19	6,402,373,705,728,000
11	3,628,800	20	121,645,100,408,832,000

- **Example 37:** Determine whether the given graph has a Hamilton circuit and path. If it does, find such a circuit and path . If it does not, give an argument to show why no such circuit and path exists.



- **Travelling Salesman Problem(TSP)**
  - The traveling salesman problem consists of a salesman and a set of cities. The salesman has to visit each one of the cities starting from a certain one (e.g. the hometown) and returning to the same city.
  - The challenge of the problem is that the traveling salesman wants to minimize the total length of the trip.
  - The goal in solving a TSP is to find the minimum cost tour, the optimal tour.
  - A tour of the vertices of a graph which visits each vertex (repeating no edge) once and only once is known as a Hamiltonian circuit.
  - Thus, one can think of solving a TSP as finding a minimum cost Hamiltonian circuit in a complete graph with weights on the edges.
  - For the general complete graph with n vertices, the number of different TSP routes would be:

$$\mathbf{(n-1)!}$$

  - However for a large value of n, this is highly inefficient algorithm.

- **Brute Froce Algorithm:**

  Optimal but inefficient algorithm when n is large. (n is number of vertices)

  A tour of the vertices of a graph which visits each vertex (repeating no edge) once.

- **Nearest-Neighour Method**

  Not-Optimal(sometime not give best answer) but efficient algorithm.

  The **nearest neighbour algorithm** was one of the first algorithms used to determine a solution to the travelling salesman problem.

  In it, the salesman starts at a random city and repeatedly visits the nearest city until all have been visited.

  It quickly yields a short tour, but usually not the optimal one.

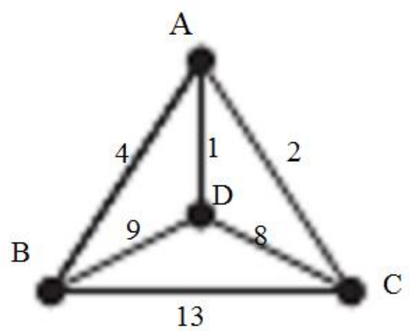
  Step of Nearest Neighbour algorithm

  1. Start on an arbitrary vertex as current vertex.
  2. Find out the shortest edge connecting current vertex and an unvisited vertex V.
  3. Set current vertex to V.
  4. Mark V as visited.
  5. If all the vertices in domain are visited, then terminate.
  6. Go to step 2.

  The sequence of the visited vertices is the output of the algorithm.

  The nearest neighbour algorithm is easy to implement and executes quickly, but it can sometimes miss shorter routes which are easily noticed with human insight, due to its "greedy" nature.

- **Example 38:**



Using Brute Force Algorithm	Using Nearest-Neighour Method
ABCDA = 4 + 13 +13 + 1 = 26	ADCBA = 26
ABDCA = 4 + 9 + 8 + 2 = 23	BADCB = 26
ACBDA = 2 + 13 + 9 + 1 = 25	CADBC = 25
	DACBA = 25

- **Example 39:** Solve the traveling salesperson problem for this graph by finding the total weight of all Hamilton circuits and determining a circuit with minimum total weight.

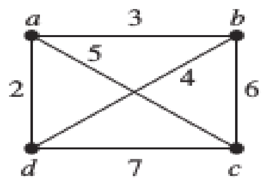


Fig a

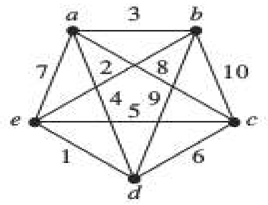


Fig b

Solution a: The following table shows the three different Hamilton circuits and their weights for fig a:

Circuit	Weight
a-b-c-d-a	3 + 6 + 7 + 2 = 18
a-b-d-c-a	3 + 4 + 7 + 5 = 19
a-c-b-d-a	5 + 6 + 4 + 2 = 17

Thus we see that the circuit a-c-b-d-a (or the same circuit starting at some other point but traversing the vertices in the same or exactly opposite order) is the one with minimu total weight.

❖ Shortest Path Problem

- In graph theory, the shortest path problem is the problem of finding a path between two vertices (or nodes) in a graph such that the sum of the weights of its constituent edges is minimized.
- The problem of finding the shortest path between two intersections on a road map (the graph's vertices correspond to intersections and the edges correspond to road segments, each weighted by the length of its road segment) may be modeled by a special case of the shortest path problem in graphs.
- There are several different algorithms that find a shortest path between two vertices in a weighted graph.

❖ Dijkstra's Algorithm

- Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph, which may represent, for example, road networks. It was conceived by computer scientist Edsger W. Dijkstra in 1956 and published three years later.

❖ Algorithm Step:

- Dijkshtra's algorithm to find the shortest path from vertex a to z of a graph G. Let G(V,E) be a simple graph and a,z $\in$ V.

- Suppose L(x) is the label of the vertex which represents the length of the shortest path from vertex a.W_{ij}=Weight of an edge $e_{ij}=(v_i,v_j)$.
- Consider following Steps:

1. Let P be the set of those vertices which have permanent labels and T be set of all vertices of G.

   Set L(a) = 0, L(x) = ∞ ∀ x ∈ T and x ≠ a

   P = ∅ and T = v.

2. Select the vertx v in T which has smallest label.This label is called the permanent label of v. Also set P as **P U {v}** and **T – {v}**

   If **v = z** then L(z) is the length of the shortest path from the vertex a to z and stop the procedure.

3. If **v ≠ z,** then revise the labels of the vertices of T. i.e. The vertices which do not have permanent labels.

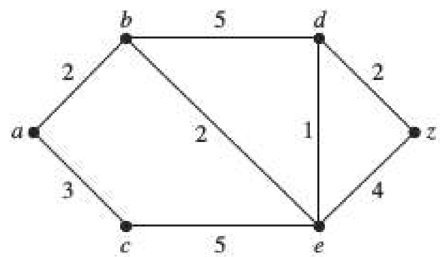
   The new label of x in T is given by

   L(x) = min{old L(x), L(v) + w(v,x)}

   Where w(v,x) is the weight of the edge joining v and x. If there is no edge joining v and x then take **w(v,x)= ∞.**

4. Repeat the step 2 and 3 until z gets permanent label.

- **Example 40:** Use Dijkstra's algorithm to find a shortest path between a and z



Solution:

Step 1: P = ∅ and T = {a,b,c,d,e,z}

L{a} = 0 L {x} = ∞

Step 2: v = a the permanent label of a = 0 P={a} T={b,c,d,e,z}

L(b) = min{old L(b), L(a) + w(a,b)} = min{∞, 0 + 2} = 2

L(c) = min{old L(c), L(a) + w(a,c)} = min{∞, 0 + 3} = 3

L(d) = min{old L(d), L(a) + w(a,d)} = min{∞, 0 + ∞} = ∞

L(e) = min{old L(e), L(a) + w(a,e)} = min{∞, 0 + ∞} = ∞

L(z) = min{old L(z), L(a) + w(a,z)} = min{∞, 0 + ∞} = ∞

Therefore L(b) = 2 is minimum label.

Step 3: v = b the permanent label of b = 2 P = {a,b} T = {c,d,e,z}

L(c) = min{old L(c), L(b) + w(b,c)} = min{3, 2 + ∞} = 3

L(d) = min{old L(d), L(b) + w(b,d)} = min{∞, 2 + 5} = 7

L(e) = min{old L(e), L(b) + w(b,e)} = min{∞, 2 + 2} = 4

L(z) = min{old L(z), L(b) + w(b,z)} = min{∞, 2 + ∞} = ∞

Therefore L(c) = 3 is minimum label.

Step 4: v = c the permanent label of c = 3 P = {a,b,c} T = {d,e,z}

L(d) = min{old L(d), L(c) + w(c,d)} = min{7, 3 +∞} = 7

L(e) = min{old L(e), L(c) + w(c,e)} = min{4, 3 + 5} = 4

L(z) = min{old L(z), L(c) + w(c,z)} = min{∞, 3 + ∞} = ∞

No labels are changed. Then e is put into P.

Therefore L(e) = 4 is minimum label.

Step 5: v = e the permanent label of e = 4 P = {a,b,c,e} T = {d,z}

L(d) = min{old L(d), L(e) + w(e,d)} = min{7, 4 +1} = 5

L(z) = min{old L(z), L(e) + w(e,z)} = min{∞, 4 + 4} = 8

Therefore L(d) = 5 is minimum label.

Step 6: v = d the permanent label of d = 5 P = {a,b,c,e,d} T= {z}

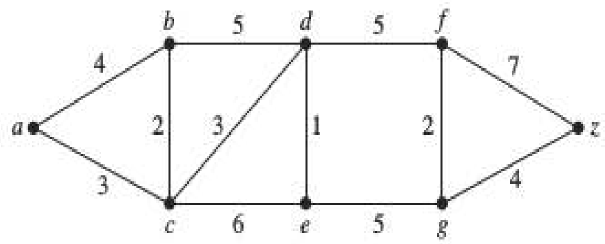
L(z) = min{old L(z), L(d) + w(d,z)} = min{8, 5 + 2} = 7

Therefore L(z) = 7 is minimum label.

Step 7: v = z, the permanemt label of z is 7

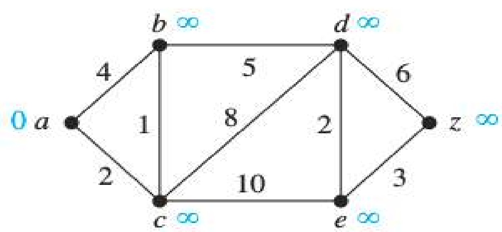
Therefore a shortest path is a, b, e, d, z, with length 7.

- **Example 41:** Use Dijkstra's algorithm to find a shortest path between a and z

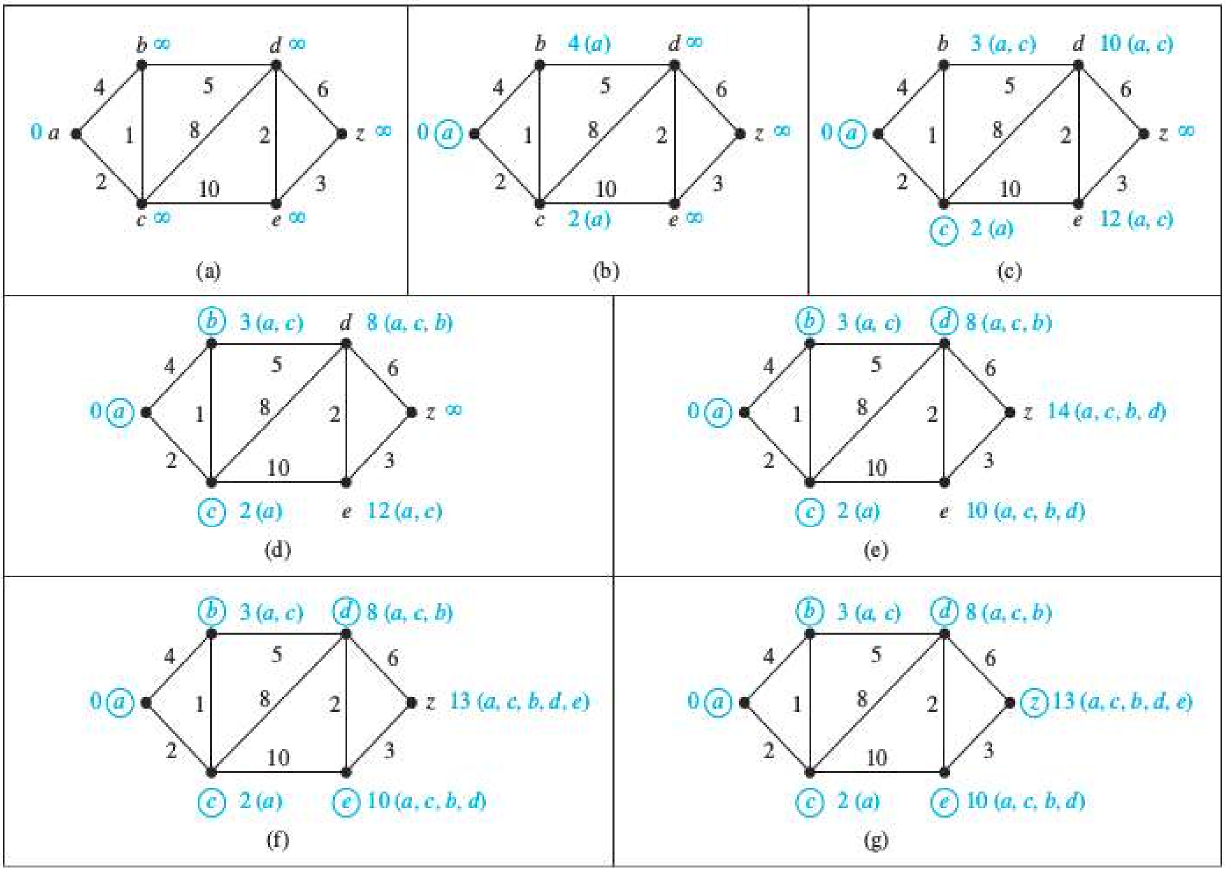


Solution: We follow the same procedure for the graph in Example 40. A shortest path is a, c, d, e, g, z, with length 16.

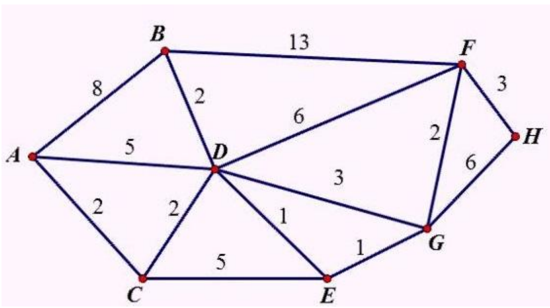
- **Example 42:** Use Dijkstra's algorithm to find the length of a shortest path between the vertices a and z in the weighted graph displayed in Figure.

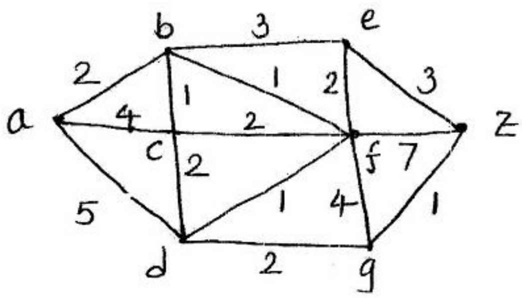


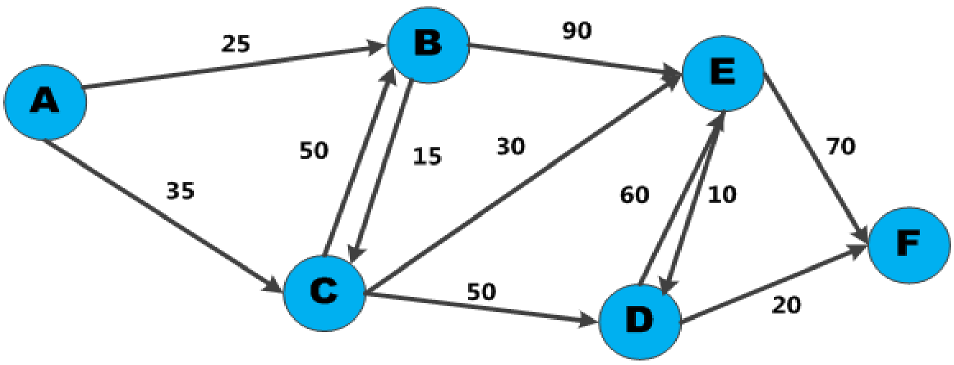
Solution:



- **Example 43:** Use Dijkstra's algorithm to find a shortest path between for following graphs.



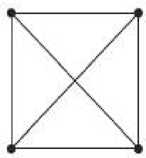




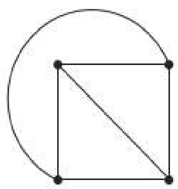
❖ Planar Graphs

- A graph is called planar if it can be drawn in the plane without any edges crossing (where a crossing of edges is the intersection of the lines or arcs representing them at a point other than their common endpoint). Such a drawing is called a planar representation of the graph.
- A graph may be planar even if it is usually drawn with crossings, because it may be possible to draw it in a different way without crossings.
- We can show that a graph is planar by displaying a planar representation. It is harder to show that a graph is nonplanar.
- **Example 44:** Is K_4 and Q_3 (shown in Figure a and c) planar?

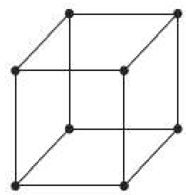
  Solution: K_4 and Q_3 are planar because it can be drawn without crossings, as shown in Figure b and d.



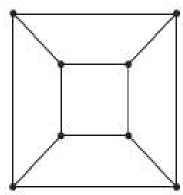
a) The Graph K_4



b) K_4 Drawn with No Crossings

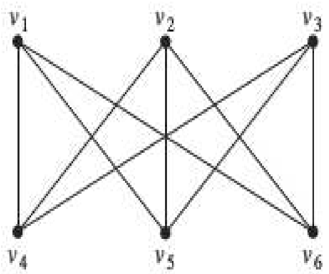


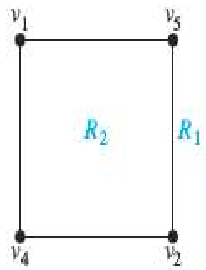
c) The Graph Q_3



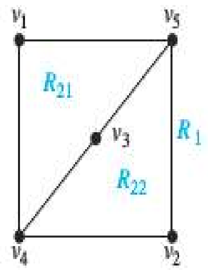
d) A Planar Representation of Q_3

- **Example 45:** Is K_{33} Planar?





(a)



(b)

Solution: Any attempt to draw $K_{3,3}$ in the plane with no edges crossing is doomed. We now show why. In any planar representation of $K_{3,3}$, the vertices v_1 and v_2 must be connected to both v_4 and v_5 .These four edges form a closed curve that splits the plane into two regions, R_1 and R_2, as shown in Fig (a). The vertex v_3 is in either R_1 or R_2. When v_3 is in R_2, the inside of the closed curve, the edges between v_3 and v_4 and between v_3 and v_5 separate R_2 into two sub regions, R_{21} and R_{22}, as shown in Fig (b). Next, note that there is no way to place the final vertex v_6 without forcing a crossing. For if v_6 is in R_1, then the edge between v_6 and v_3 cannot be drawn without a crossing. If v_6 is in R_{21}, then the edge between v_2 and v_6 cannot be drawn without a crossing. If v_6 is in R_{22}, then the edge between v_1 and v_6 cannot be drawn without a crossing.

A similar argument can be used when v_3 is in R_1. It follows that $K_{3,3}$ is not planar.

❖ Euler's Formula

- Euler showed that all planar representations of a graph split the plane into the same number of regions. He accomplished this by finding a relationship among the number of regions, the number of vertices, and the number of edges of a planar graph.
- Let G be a connected planar simple graph with e edges and v vertices. Let r be the number of regions in a planar representation of G. **Then $r = e - v + 2$.**
- **Example 46:** Suppose that a connected planar graph has 20 vertices, each of degree 3. Into how many regions does a representation of this planar graph split the plane?
  Solution: This graph has 20 vertices, each of degree 3, so $v = 20$.
  Because the sum of the degrees of the vertices, $3v = 3 * 20 = 60$.
  By Handshaking Lemma In a graph, the sum of all the degrees of vertices is equal to twice the number of edges. Therefore we have $2e = 60$, or $e = 30$.
  From Euler's formula, the number of regions is $r = e - v + 2 = 30 - 20 + 2 = 12$.

- **Corollary 1:** If G is a connected planar simple graph with e edges and v vertices, where $v \geq 3$, then $e \leq 3v - 6$.
- **Corollary 2:** If G is a connected planar simple graph, then G has a vertex of degree not exceeding five.
- **Corollary 3:** If a connected planar simple graph has e edges and v vertices with $v \geq 3$ and no circuits of length three, then $e \leq 2v - 4$.

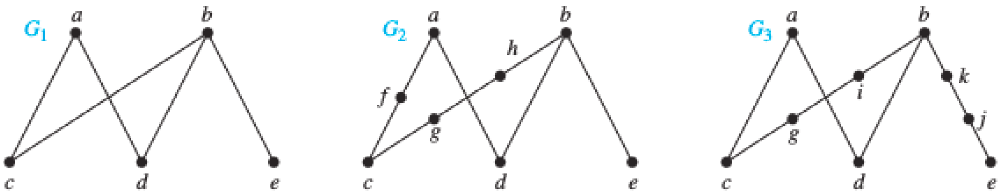
- **Example 47:** Show that K_5 is nonplanar using Corollary 1.
  Solution: The graph K_5 has 5 vertices and 10 edges. However, the inequality $e \leq 3v - 6$ is not satisfied for this graph because $e = 10$ and $3v - 6 = 9$. Therefore, K_5 is not planar.

  It was previously shown that$K_{3,3}$ is not planar. Note, however, that this graph has six vertices and nine edges. This means that the inequality $e=9 \leq 12 = 3*6 - 6$ is satisfied. Consequently, the fact that the inequality $e \leq 3v-6$ is satisfied does not imply that a graph is planar. However, the corollary 3 can be used to show that $K_{3,3}$ is nonplanar.

- **Example 48:** Use Corollary 3 to show that $K_{3,3}$ is nonplanar.
  Solution: Because $K_{3,3}$ has no circuits of length three (this is easy to see because it is bipartite), Corollary 3 can be used. $K_{3,3}$ has six vertices and nine edges. Because $e = 9$ and $2v - 4 = 8$, Corollary 3 shows that $K_{3,3}$ is nonplanar.

❖ Kuratowski's Theorem

- We have seen that $K_{3,3}$ and K_5 are not planar. Clearly, a graph is not planar if it contains either of these two graphs as a subgraph. Surprisingly, all nonplanar graphs must contain a subgraph that can be obtained from $K_{3,3}$ or K_5 using certain permitted operations.
- If a graph is planar, so will be any graph obtained by removing an edge {u, v} and adding a new vertex w together with edges {u, w} and {w, v}.
- Such an operation is called an **elementary subdivision.**
- The graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are called **homeomorphic** if they can be obtained from the same graph by a sequence of elementary subdivisions.
- **Theorem:** A graph is nonplanar if and only if it contains a subgraph homeomorphic to $K_{3,3}$ or K_5.
- **Example 49:** Show that the graphs G_1, G_2 and G_3 displayed in Figure below are all homeomorphic.



Solution: These three graphs are homeomorphic because all three can be obtained from G_1 by elementary subdivisions. G_1 can be obtained from itself by an empty sequence of elementary subdivisions.

To obtain G_2 from G_1 we can use this sequence of elementary subdivisions:

(i) remove the edge {a, c}, add the vertex f , and add the edges {a, f } and {f, c};

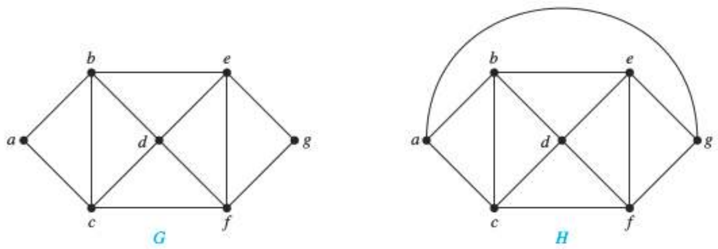
(ii) remove the edge {b, c}, add the vertex g, and add the edges {b, g} and {g, c}; and

(iii) remove the edge {b, g}, add the vertex h, and add the edges {g, h} and {b, h}.

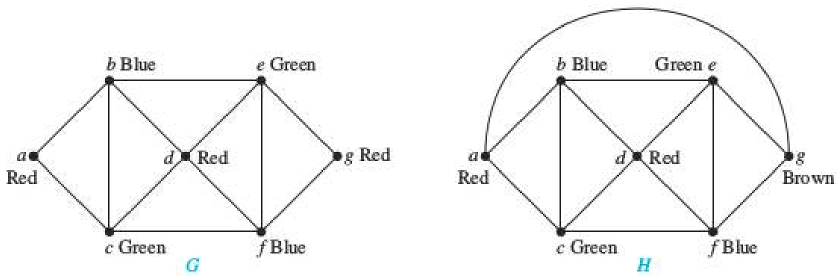
Similarly for to obtain G_3 from G_1. Repeat the above steps.

❖ Graph Coloring

- A **coloring of a simple graph** is the assignment of a color to each vertex of the graph so that no two adjacent vertices are assigned the same color.
- The **chromatic number of a graph** is the least number of colors needed for a coloring of this graph. The chromatic number of a graph G is denoted by $\chi(G)$.
- **The Four Color Theorem:** The chromatic number of a planar graph is no greater than four.
- **Example 50:** What are the chromatic numbers of the graphs G and H in Figure?



Solution: As show in figure below Coloring of Graph G has chromatic number equal to 3 and Graph H has a chromatic number equal to 4.



- **Example 51:** What is the chromatic number of K_n?

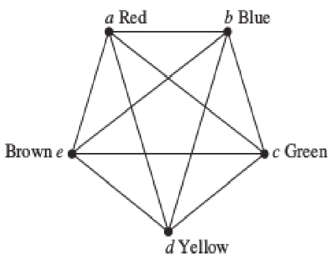
  Solution: A coloring of K_n can be constructed using n colors by assigning a different color to each vertex. The chromatic number of K_n is n. That is, $\chi(K_n) = n$.

- **Example 52**: What is the chromatic number of the complete bipartite graph $K_{m,n}$ where m and n are positive integers?

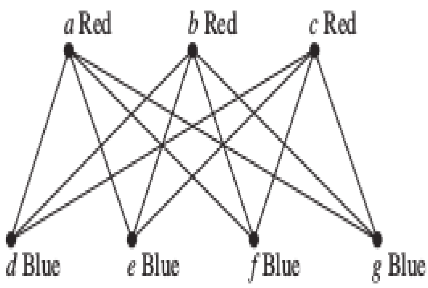
  Solution: The number of colors needed may seem to depend on m and n. $K_{m,n}$ is a bipartite graph. The chromatic number of $K_{m,n}$ is 2. That is $\chi(K_{m,n}) = 2$.

- **Example 53**: What is the chromatic number of the graph C_n where $n \geq 3$?

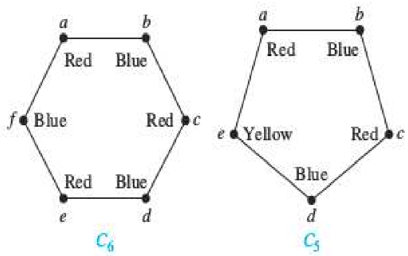
  Solution: In general, the chromatic number of C_n is 2 when n is even, and the chromatic number of C_n is 3 when n is odd and $n > 1$.



Coloring of K_5



Coloring of $K_{3,4}$



Colorings of C_5 and C_6