Discrete Mathematics (ITPC-309)

Graphs – Part II

Mrs. Sanga G. Chaki

Department of Information Technology

Dr. B. R. Ambedkar National Institute of Technology, Jalandhar

Contents

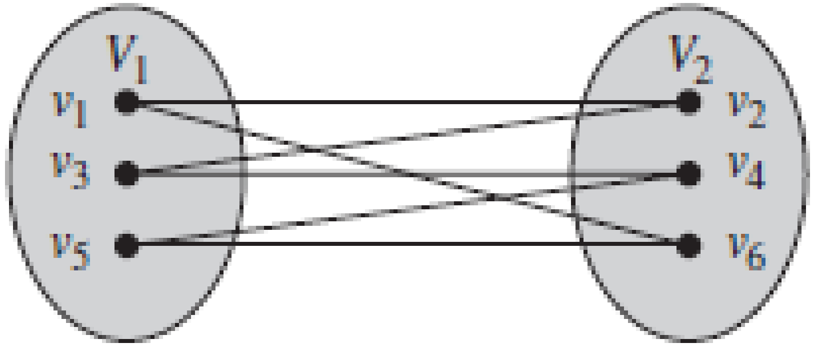
- What are graphs
- Applications
- Basic Terminology
- Finite and infinite graph
- Simple graph, multi graph,
- graph terminology,
- representation of graphs,
- Bipartite, Regular, Planar and connected graphs,
- Connected components in a graph,
- Euler graphs,
- Hamiltonian path and circuits,
- Graph coloring, chromatic number,
- Isomorphism and Homomorphism of graphs.

Bipartite Graphs

Bipartite Graphs

1. A simple graph G is called bipartite if its vertex set V can be partitioned into two disjoint sets V_1 and V_2 such that every edge in the graph connects a vertex in V_1 and a vertex in V_2 and
2. No edge in G connects either two vertices in V_1 or two vertices in V_2.
3. When this condition holds, we call the pair (V_1, V_2) a **bipartition of the vertex set V** of G.

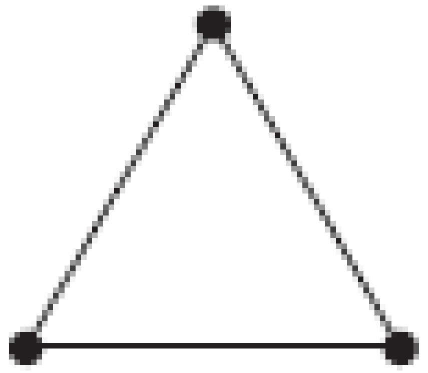
Bipartite Graphs



1. Example: Below graph is bipartite, because its vertex set can be partitioned into the two sets
2. V_1 = {v1, v3, v5} and V_2 = {v2, v4, v6},
3. and every edge of connects a vertex in V1 and a vertex in V2.
4. **This is C6 graph – cycle with 6 vertices**

Bipartite Graphs

1. Is the following bipartite?

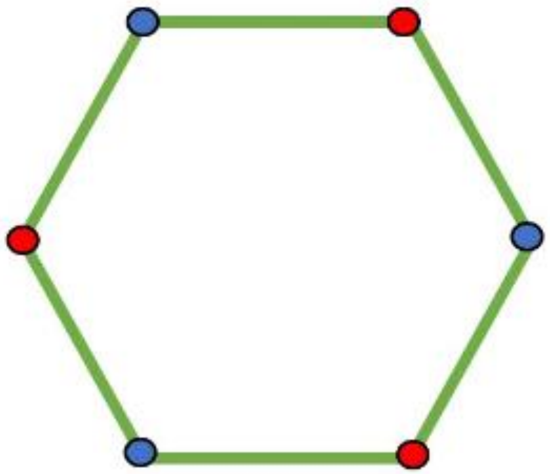


K_3

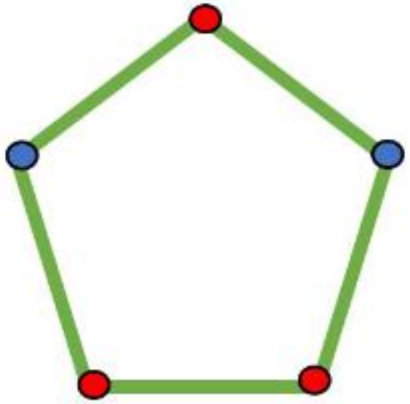
2. K3 is not bipartite.
3. To verify this, note that if we divide the vertex set of K3 into two disjoint sets, one of the two sets must contain two vertices.
4. If the graph were bipartite, these two vertices could not be connected by an edge, but in K3 each vertex is connected to every other vertex by an edge.

Bipartite Graphs

1. How to determine whether a graph is bipartite?
2. 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. Eg. C_6
3. It is not possible to color a cycle graph with odd cycle using two colors. C_5
4. This is an example from graph colorings – we will study later.



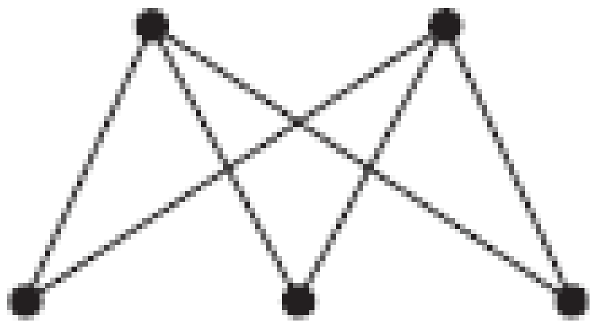
Cycle graph of length 6



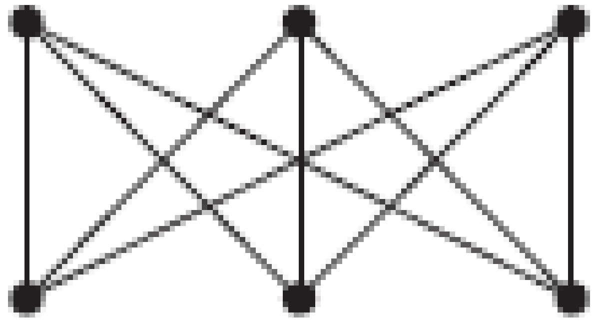
Cycle graph of length 5

Bipartite Graphs

1. **Complete Bipartite Graphs**:
2. 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
3. where every vertex of the first set is connected to every vertex of the second set



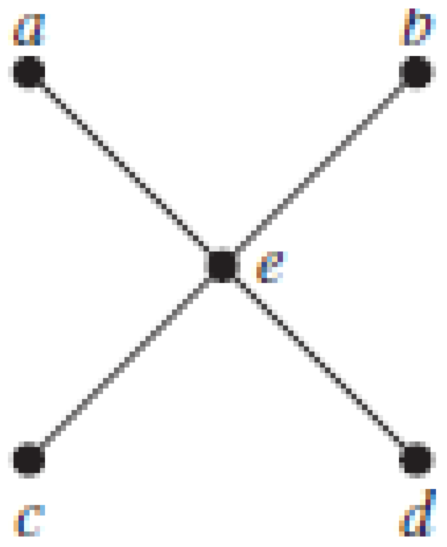
$K_{2,3}$

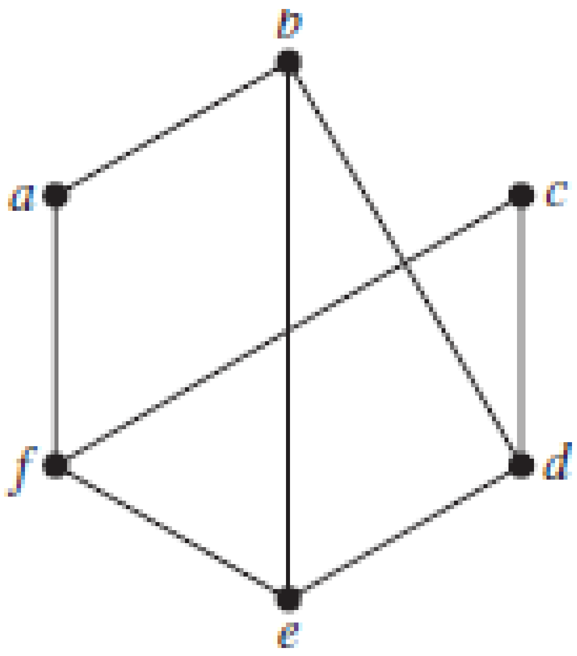


$K_{3,3}$

Bipartite Graphs

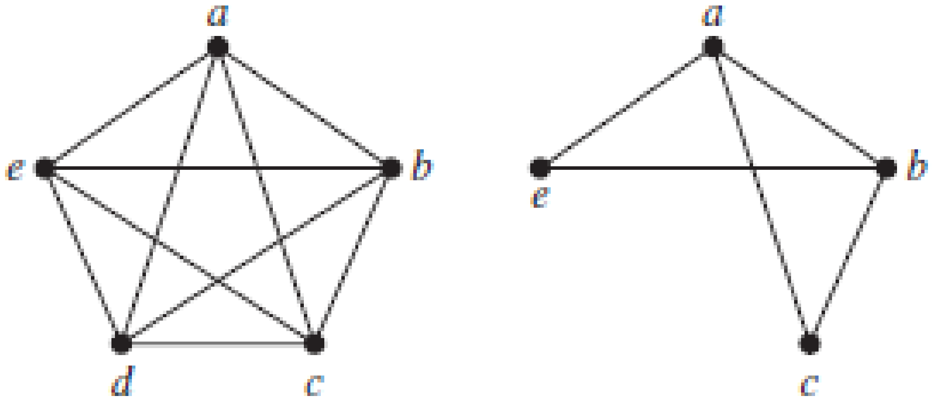
1. Determine if the following are bipartite?





SubGraphs

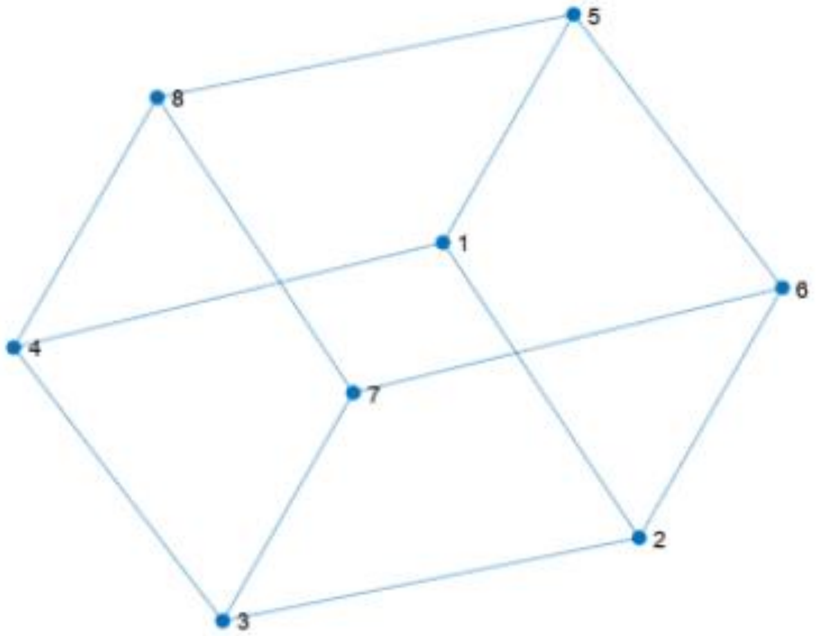
Subgraphs

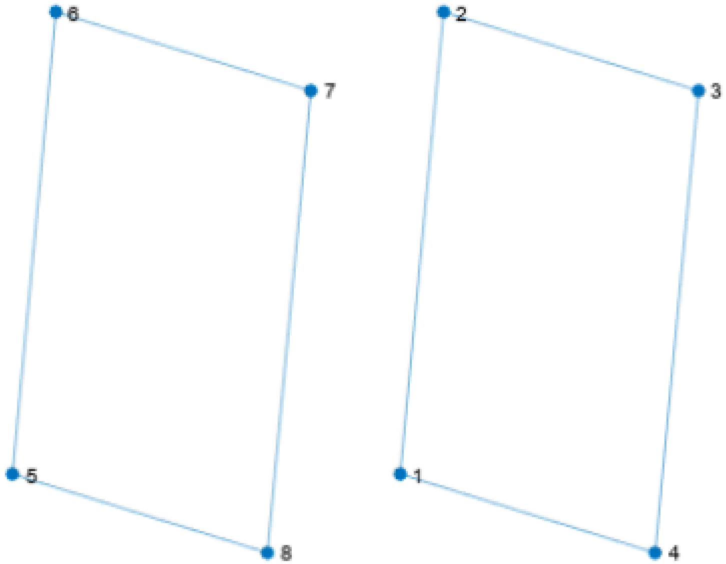


1. A subgraph of a graph G = (V ,E) is a graph H = (W, F), where W ⊆ V and F ⊆ E.
2. A subgraph H of G is a proper subgraph of G if H ≠ G.
3. In the subgraph H, the edge set F contains an edge from E if and only if both endpoints of this edge are in W.
4. The graph G shown in this figure is a subgraph of K5.

Removing Edges Of A Graph

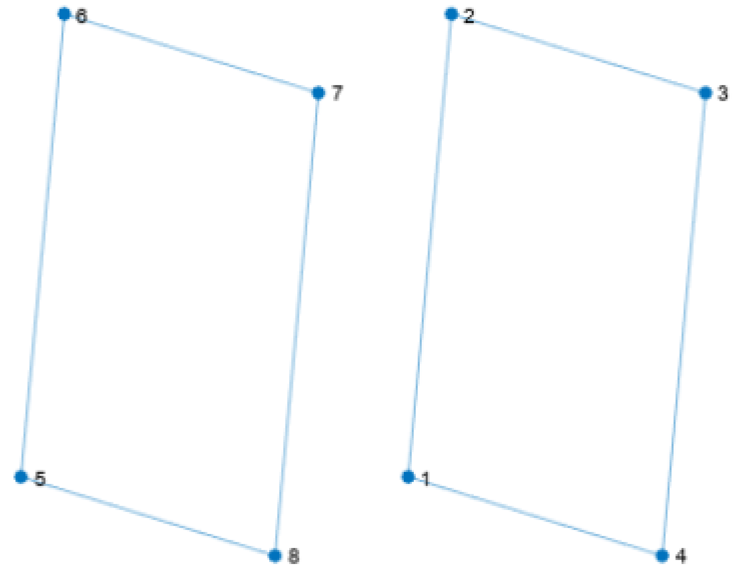
1. Given a graph G = (V ,E) and an edge e ∈ E, we can produce a subgraph of G by removing the edge e.
2. The resulting subgraph, denoted by G − e, has the same vertex set V as G.
3. Its edge set is E − e.
4. Hence, G − e = (V ,E − {e}).

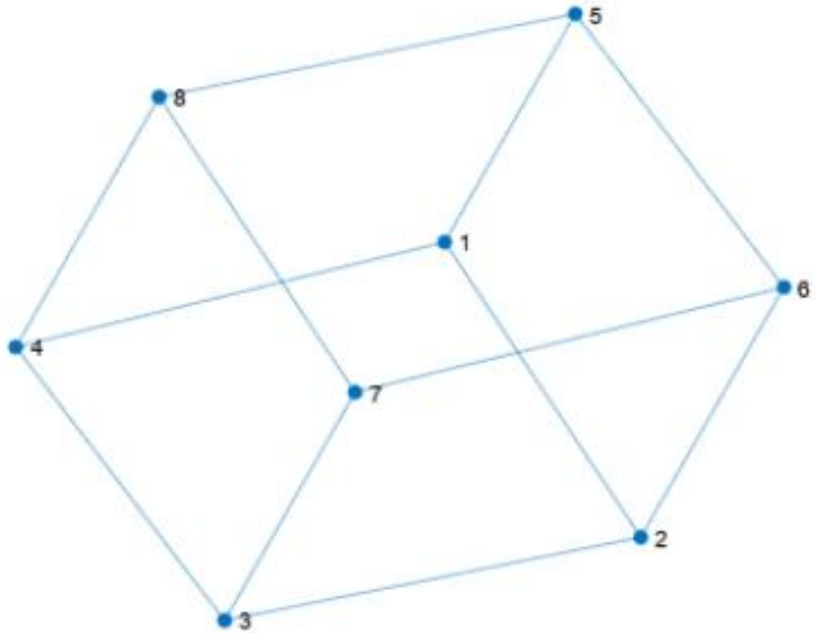




Adding Edges To A Graph

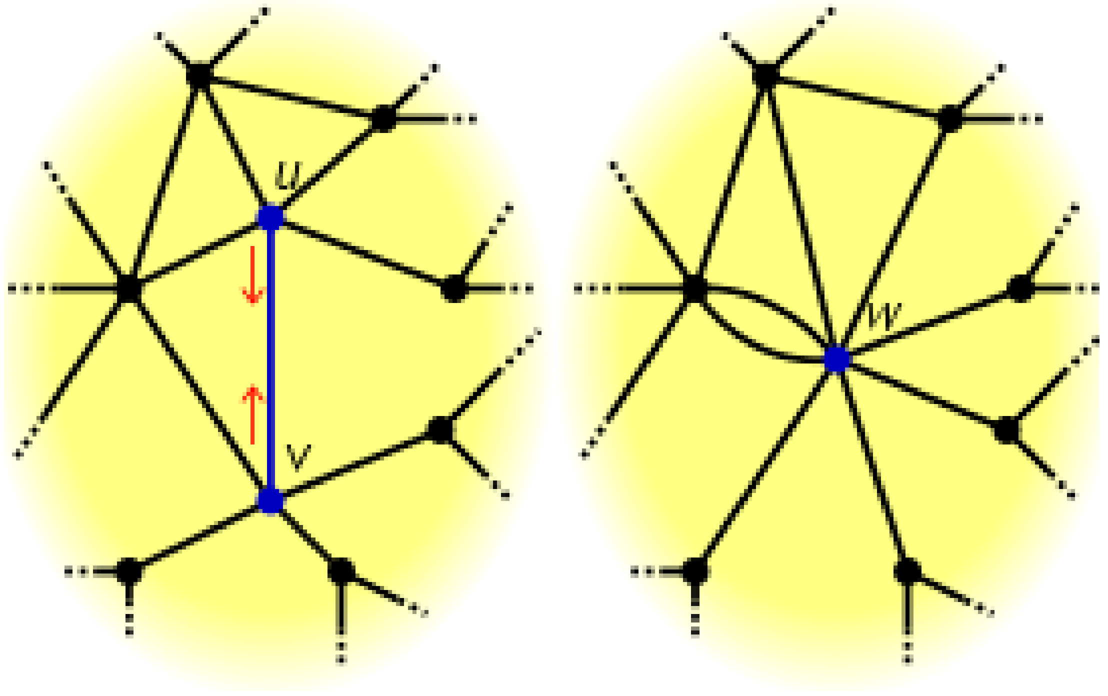
1. We can also add an edge e to a graph to produce a new larger graph
2. This edge must connect two vertices already in G.
3. We denote by G + e the new graph produced by adding a new edge e, connecting two previously non-incident vertices, to the graph G.
4. Hence, G + e = (V ,E ∪ {e}).
5. The vertex 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}.





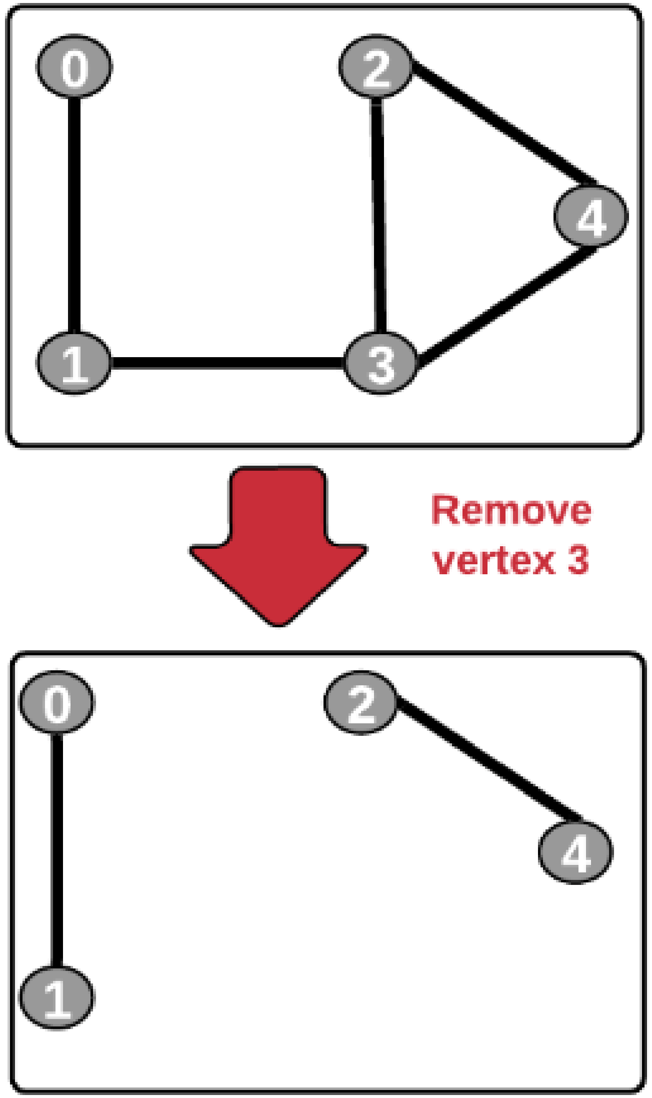
Edge Contractions

1. Used when we want to remove an edge from a graph, and its endpoints
2. Edge contraction removes an edge e with endpoints u and v and merges u and v into a new single vertex w.
3. For each edge with u or v as an endpoint replace the edge with one with w as endpoint in place of u or v and with the same second endpoint
4. This new graph is not a subgraph of G



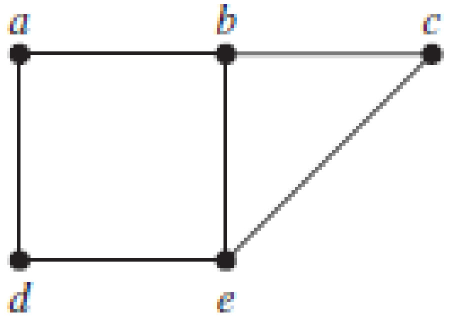
Removing Vertices From A Graph

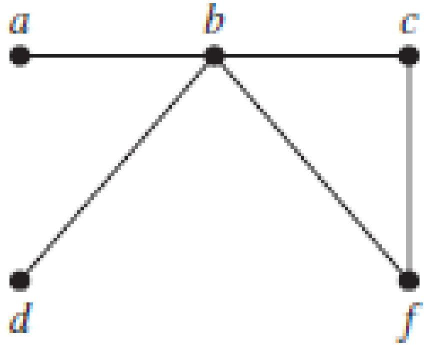
1. When we remove a vertex v and all edges incident to it from G = (V ,E), we produce a subgraph, denoted by G – v
2. G – v = (V – v, E’), where E’ is the set of edges of G not incident to v.

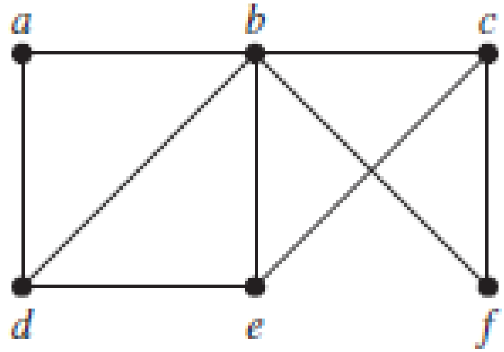


Graph Unions

1. Graph Unions: 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.
2. Example: Find the union of the graphs G1 and G2

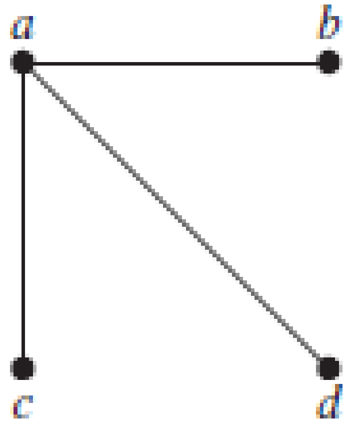




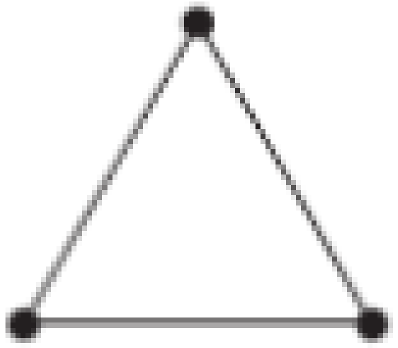


Examples

1. Draw all subgraphs of this graph.



2. How many subgraphs with at least one vertex does K3 (complete graph with three vertices) have?



K_3

Graph Representations

1. Adjacency lists
2. Adjacency matrices
3. Incidence matrices

Representing Graphs: Adjacency lists

1. Make a list of all the vertices in a graph
2. Specify the vertices that are adjacent to each vertex of the graph.
3. Works for graph without multiple edges
4. Example:

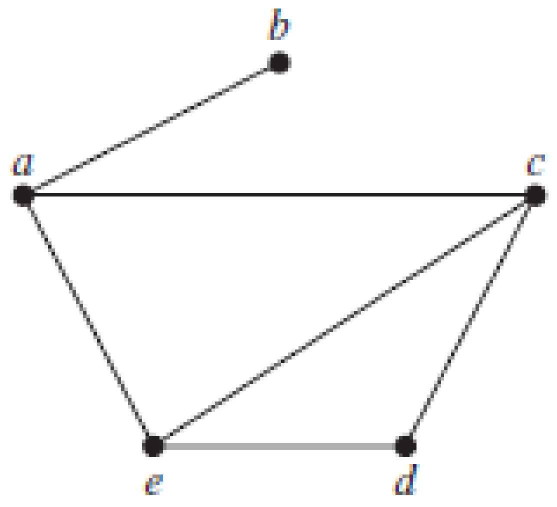


TABLE 1 **An Adjacency List for a Simple Graph.**

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

Representing Graphs: Adjacency lists

1. Example: Represent the directed graph shown in the figure by listing all the vertices that are the terminal vertices of edges starting at each vertex of the graph.

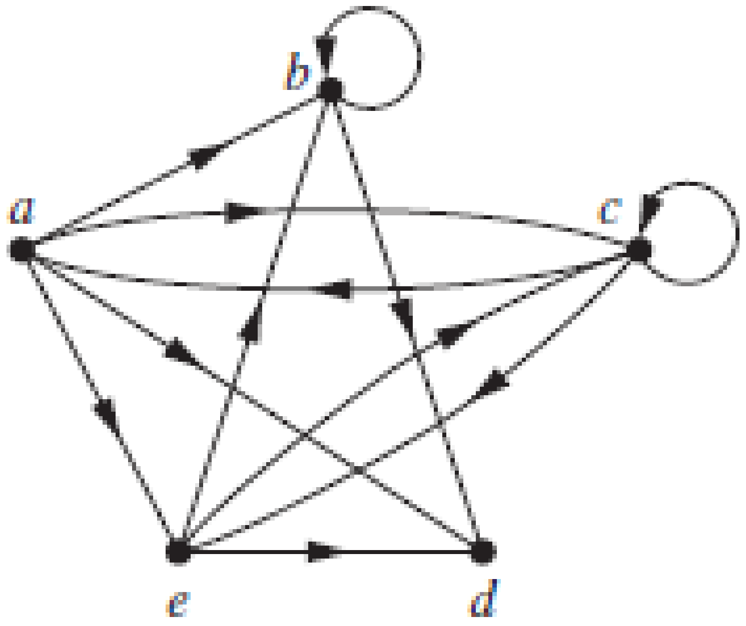


TABLE 2 An Adjacency List for a Directed Graph.

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

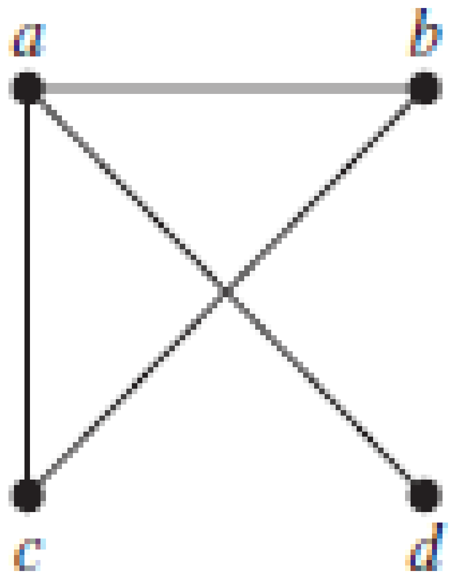
Adjacency Matrices

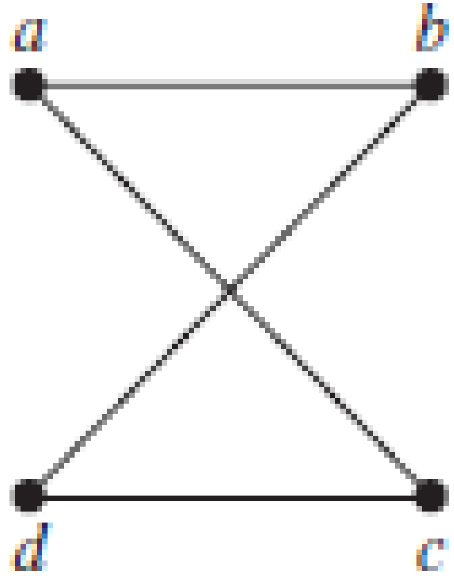
1. To simplify computation, graphs can be represented using matrices.
2. Suppose that G = (V ,E) is a simple graph where |V| = n.
3. Suppose that the vertices of G are listed as $v_1, v_2, \ldots, v_n$.
4. The adjacency matrix A (or A_G) of G, with respect to this listing of the vertices, is the **n x n zero–one matrix** A = [a_{ij}], where

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

Adjacency Matrices

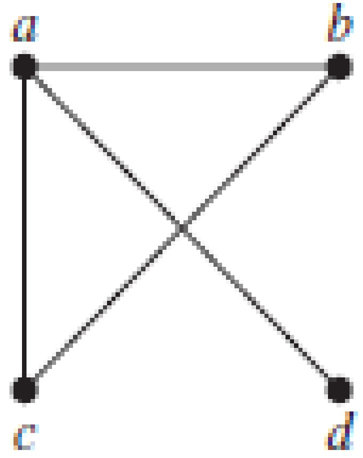
1. Use an adjacency matrix to represent the graphs

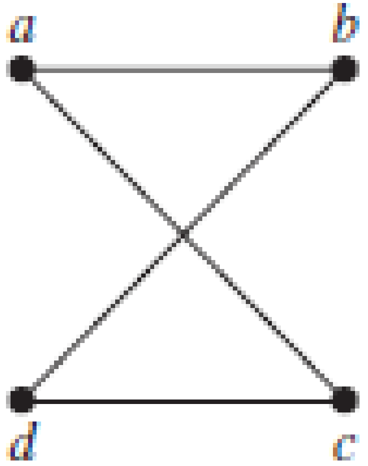




Adjacency Matrices

1. Use an adjacency matrix to represent the graphs





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

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

Adjacency Matrices – Some Properties

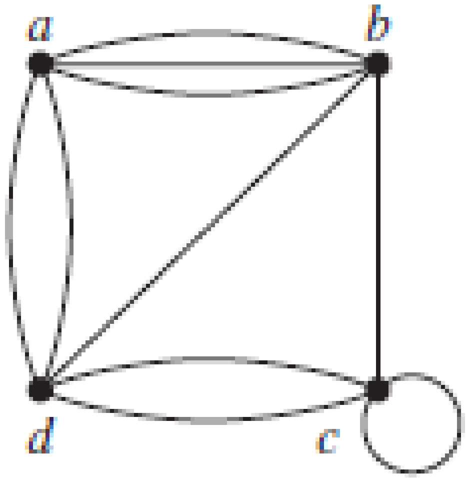
1. An adjacency matrix of a graph is based on the ordering chosen for the vertices.
   - Hence, there may be as many as n! different adjacency matrices for a graph with n vertices, because there are n! different orderings of n vertices.
2. The adjacency matrix of a simple graph is symmetric, that is, $a_{ij} = a_{ji}$,
   - because both of these entries are 1 when v_i and v_j are adjacent, and both are 0 otherwise.
3. Furthermore, because a simple graph has no loops,
   - each entry a_{ii}, $i = 1, 2, 3, \ldots, n$, is 0.

Adjacency Matrices

1. Adjacency matrices can also be used to represent undirected graphs with loops and with multiple edges
2. A loop at the vertex v_i is represented by a 1 at the $(i, i)^{th}$ position of the adjacency matrix.
3. When multiple edges or multiple loops at the same vertex, are present,
   - the $(i, j)^{th}$ entry of this matrix = the number of edges that are associated to $\{v_i, v_j\}$.

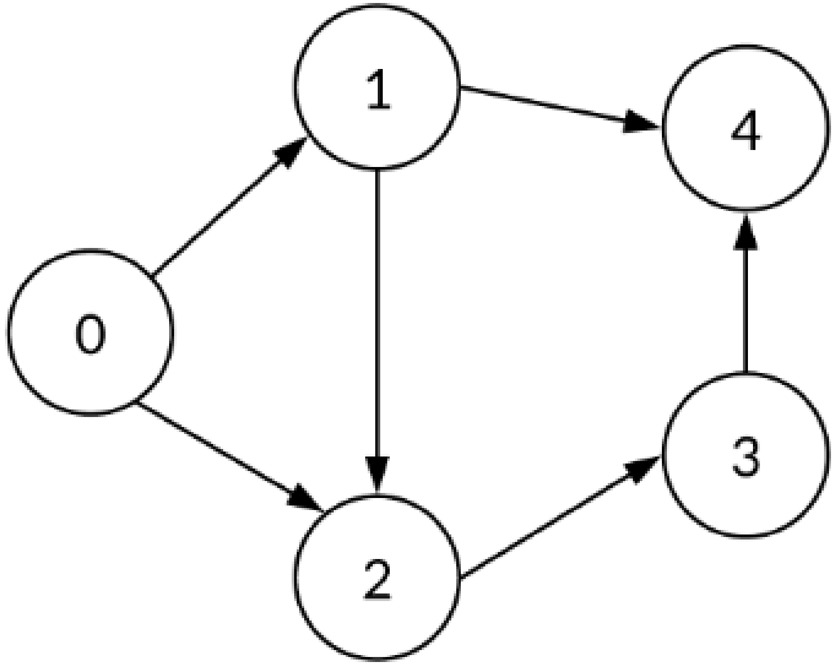
Adjacency Matrices

1. Use an adjacency matrix to represent the pseudograph shown:
2. 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



Adjacency Matrix

	0	1	2	3	4
0	0	1	1	0	0
1	0	0	1	0	1
2	0	0	0	1	0
3	0	0	0	0	1
4	0	0	0	0	0

1. For directed graphs, adjacency matrices are possible.
2. The matrix for a directed graph G = (V ,E) has a 1 in its (i, j)th position if there is an edge from v_i to v_j
3. The adjacency matrix for a directed graph does not have to be symmetric

Incidence Matrices

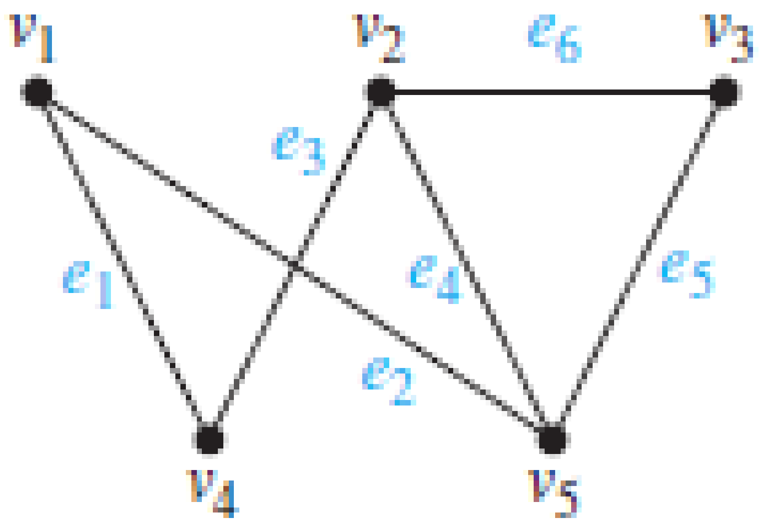
1. Let G = (V ,E) be an undirected graph.
2. Suppose that $v_1, v_2, \ldots, v_n$ are the vertices and $e_1, e_2, \ldots, e_m$ are the edges of G.
3. The incidence matrix with respect to this ordering of V and E is the $n \times m$ matrix $M = [m_{ij}]$, where

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

4. Incidence matrices can also be used to represent multiple edges and loops

Incidence Matrices

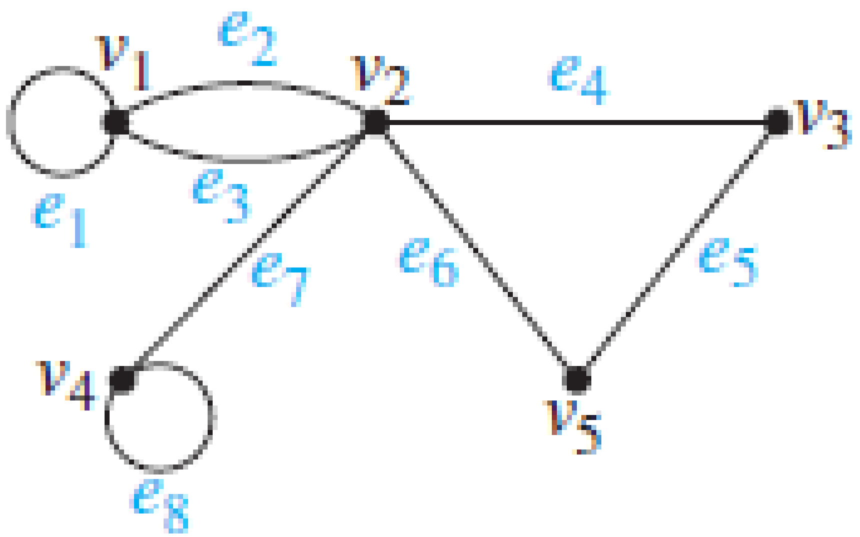
1. Example: Represent the graph shown with an incidence matrix



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

Incidence Matrices

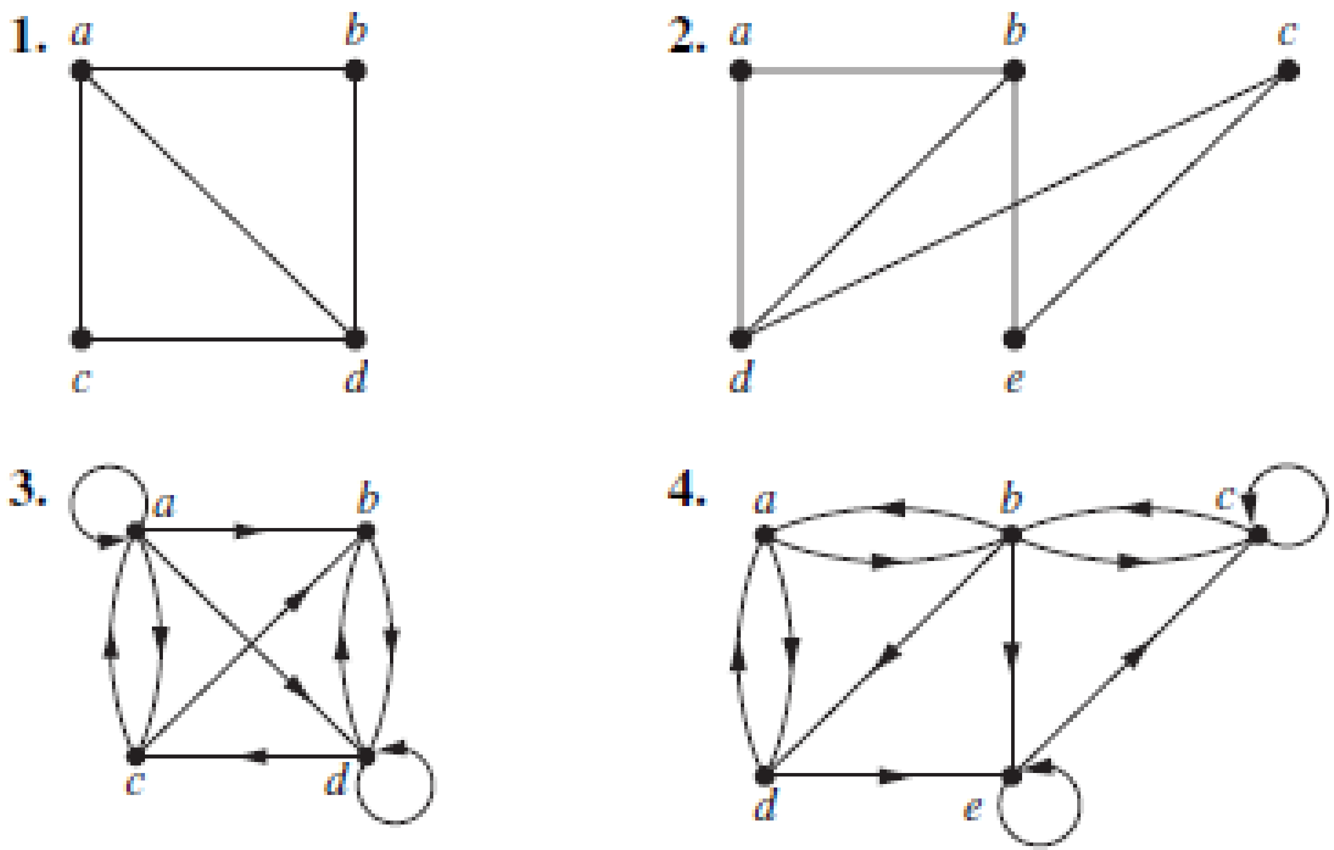
1. Example: Represent the graph shown with an incidence matrix



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

Examples

1. Use adjacency list, adjacency matrix and incidence matric to represent the given graphs.



Isomorphism in Graphs

Isomorphism of Graphs

1. A graph can exist in different forms having the same number of vertices, edges, and also the same edge connectivity.
2. Such graphs are called isomorphic graphs.
3. Two graphs G1 and G2 are said to be isomorphic if –
   - Their number of components (vertices and edges) are same.
   - Their edge connectivity is retained.

Isomorphism of Graphs - Formally

There exists a function ‘f’ from vertices of G1 to vertices of G2

[f: V(G1) ⇒ V(G2)], such that

- Case (i): f is a bijection (both one-one and onto)
- Case (ii): f preserves adjacency of vertices, i.e., if the edge {U, V} ∈ G1, then the edge {f(U), f(V)} ∈ G2,

then G1 ≡ G2, that is G1 and G2 are isomorphic

Isomorphism of Graphs

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*.

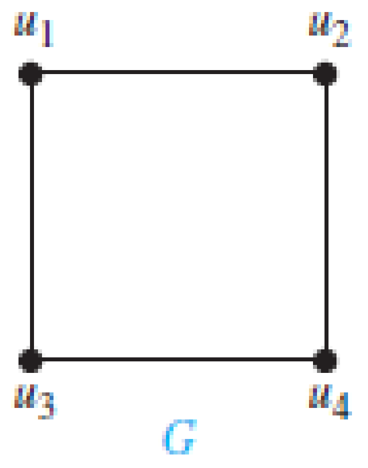
1. What this means?
2. When two simple graphs are isomorphic, there is a one-to-one correspondence between vertices of the two graphs that preserves the adjacency relationship

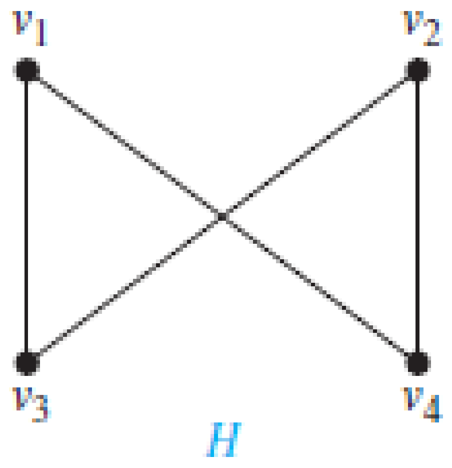
Determining whether Two Simple Graphs are Isomorphic

1. 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.
2. 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.
3. The **degrees of the vertices in isomorphic simple graphs must be the same**.
   - That is, a vertex v of degree d in G must correspond to a vertex f(v) of degree d in H, because a vertex w in G is adjacent to v if and only if f(v) and f(w) are adjacent in H.

Isomorphism of Graphs

1. Example: Show that the graphs G = (V ,E) and H = (W, F), are isomorphic

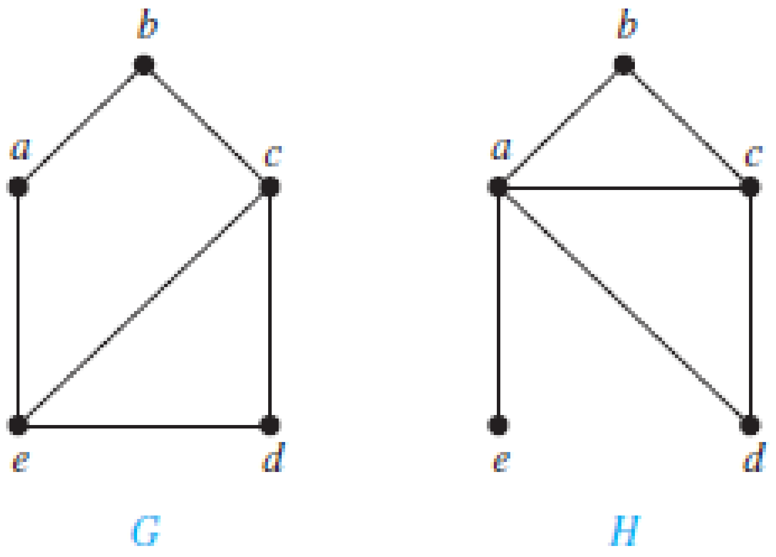




1. Number of vertices and edges are same.
2. Corresponding vertices:
   - The function f with f (u1) = v1, f (u2) = v4, f (u3) = v3, and f (u4) = v2 is a one-to-one correspondence between V and W.
3. Does this correspondence preserves adjacency? **You can use adjacency matrix**.
   - u1-u2 = v1-v4
   - u1-u3 = v1-v3
   - And so on
4. So these graphs are isomorphic.

Isomorphism of Graphs

1. Are the following graph pairs isomorphic? Prove it.



2. The problem of determining whether any two graphs are isomorphic is of special interest because it is one of only a few NP problems – nondeterministic polynomial – solution can be guessed and verified in polynomial time.