BBM 205 Discrete Mathematics
Hacettepe University
http://web.cs.hacettepe.edu.tr/∼bbm205

Lecture 7: Graph Terminology and Graph Isomorphism

Resources:
Kenneth Rosen, "Discrete Mathematics and App."
http://www.inf.ed.ac.uk/teaching/courses/dmmr
http://www.cs.nthu.edu.tw/ wkhon/math16.html

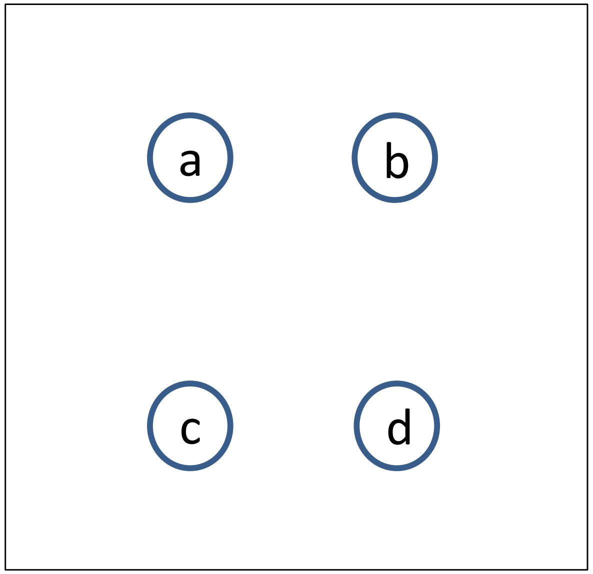
What is a Graph ?

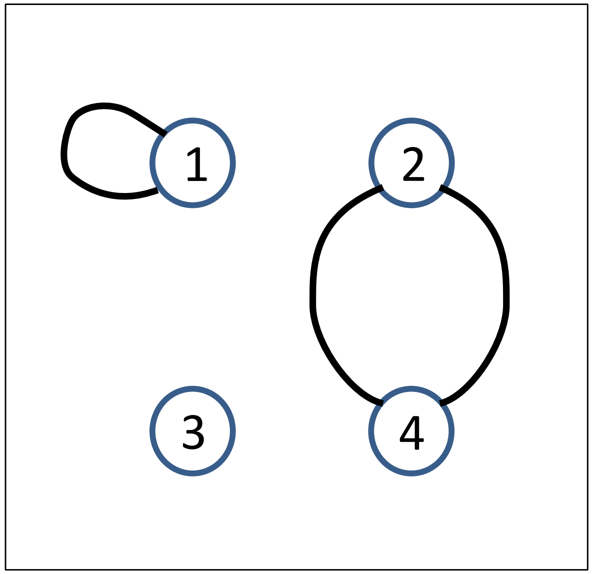
A graph consists of a nonempty set V of vertices and a set E of edges, where each edge in E connects two (may be the same) vertices in V.

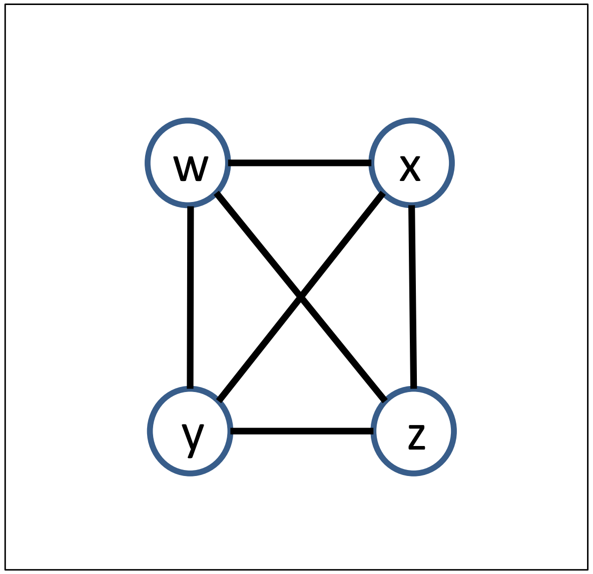
- Let G be a graph associated with a vertex set V and an edge set E

  We usually write G = (V, E) to indicate the above relationship

Examples







- Furthermore, if each edge connects two different vertices, and no two edges connect the same pair of vertices, then the graph is a simple graph
- Which of the above is a simple graph ?

Directed Graph

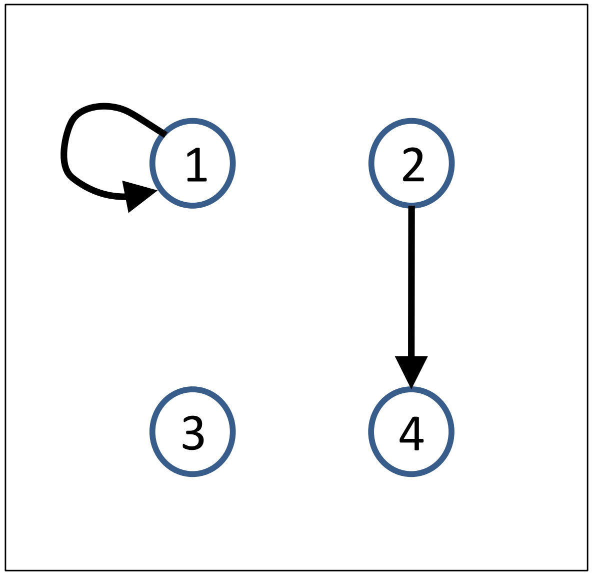
- Sometimes, we may want to specify a direction on each edge

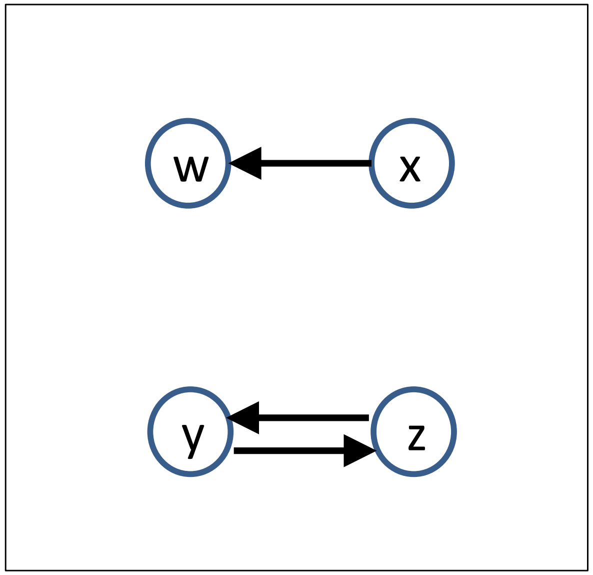
  Example : Vertices may represent cities, and edges may represent roads (can be one-way)

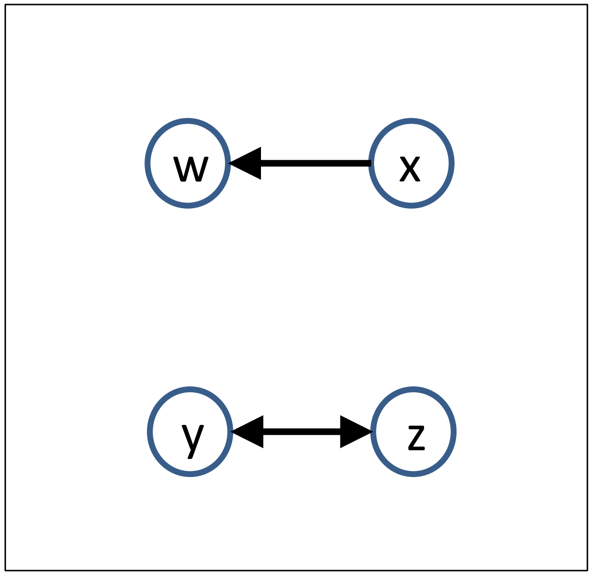
- This gives the directed graph as follows :

A directed graph G consists of a nonempty set V of vertices and a set E of directed edges, where each edge is associated with an ordered pair of vertices. We write G = (V, E) to denote the graph.

Examples







Test Your Understanding

- Suppose we have a simple graph G with n vertices

  What is the maximum number of edges G can contain, if

  (i) G is an undirected graph ?

  (ii) G is a directed graph ?

Terminology (Undirected Graph)

- Let e be an edge that connects vertices u and v

  We say (i) e is incident with u and v

  (ii) u and v are the endpoints of e ;

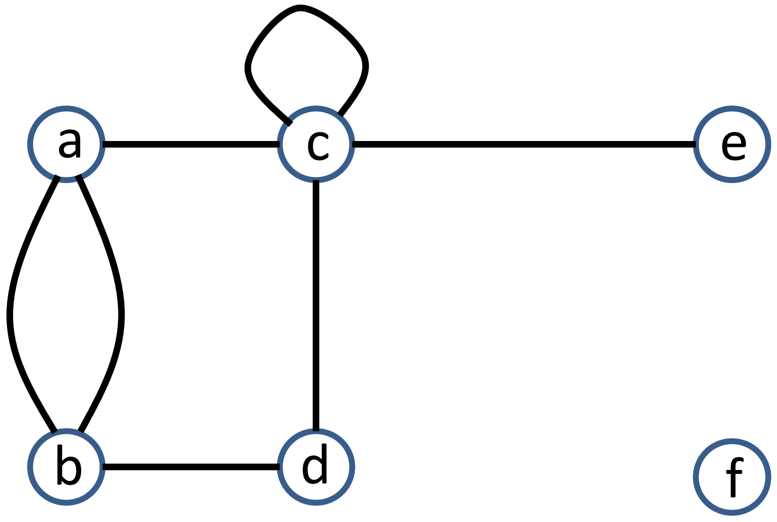
  (iii) u and v are adjacent (or neighbors)

  (iv) if u = v, the edge e is called a loop

- The degree of a vertex v, denoted by deg(v), is the number of edges incident with v, except that a loop at v contributes twice to the degree of v

Example

- What are the degrees and neighbors of each vertex in the following graph ?



Handshaking Theorem

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

Theorem:

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

- Proof : Each edge e contributes exactly twice to the sum on the left side (one to each endpoint).

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

Terminology (Directed Graph)

- Let e be an edge that connects vertices from u to v

  We say (i) u = initial vertex, v = terminal vertex ;

  (ii) u is adjacent to v;

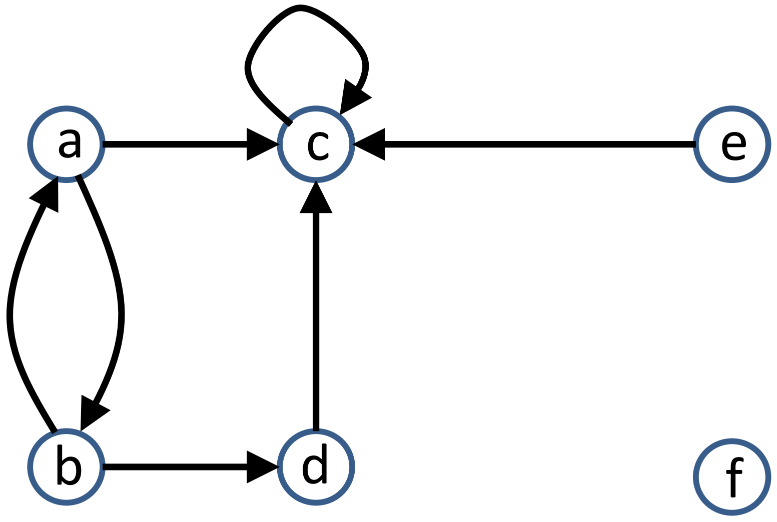
  (iii) v is adjacent from u;

  (iv) if u = v, the edge e is called a loop

- The in-degree of a vertex v, denoted by $\deg^{-}(v)$, is the number of edges with v as terminal vertex
- The out-degree of a vertex u, denoted by $\deg^{+}(u)$, is the number of edges with u as initial vertex

Example

- What are the in- and out-degrees of each vertex in the following graph ?



Handshaking Theorem

- Let G = (V, E) be directed graph with m edges

Theorem:

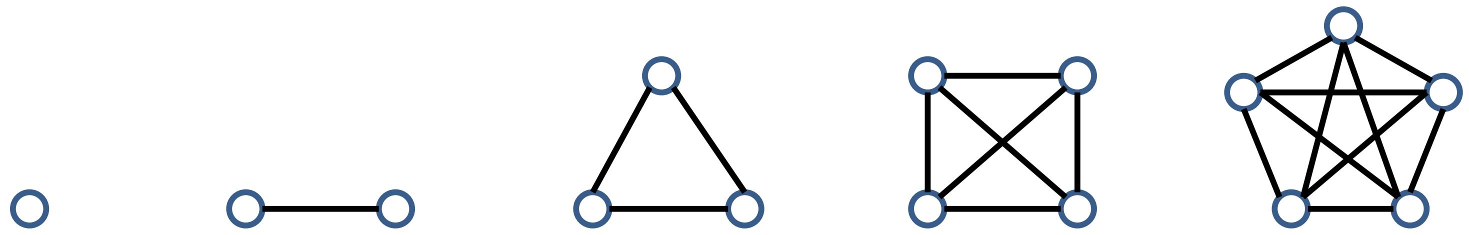
$$\sum_{v \in V} \deg^{-}(v) = \sum_{u \in V} \deg^{+}(u) = m$$

- Proof : Each edge e contributes exactly once to the in-degree and once to the out-degree

Some Special Simple Graphs

Definition: A complete graph on n vertices, denoted by K_n, is a simple graph that contains one edge between each pair of distinct vertices

Examples :

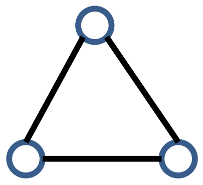


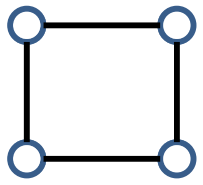
K_1 K_2 K_3 K_4 K_5

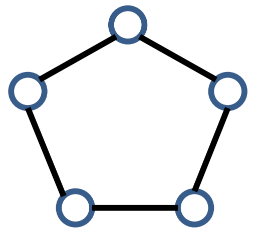
Some Special Simple Graphs

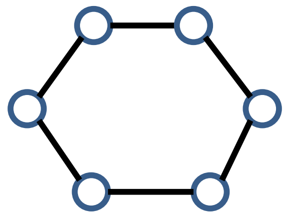
Definition: A cycle C_n, $n \geq 3$, is a graph that consists of n vertices $v_1, v_2, \ldots, v_n$ and n edges $\{v_1, v_2\}$, $\{v_2, v_3\}$, …, $\{v_{n-1}, v_n\}$, $\{v_n, v_1\}$

Examples :


C_3


C_4

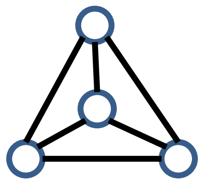

C_5


C_6

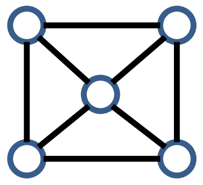
Some Special Simple Graphs

Definition: A wheel W_n, $n \geq 3$, is a graph that consists of a cycle C_n with an extra vertex that connects to each vertex in C_n

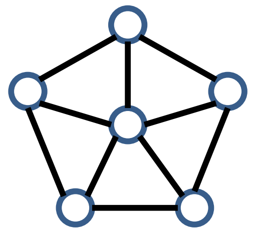
Examples :



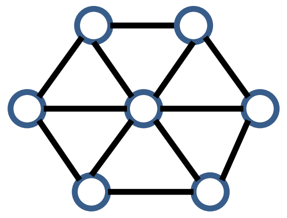
W_3



W_4



W_5

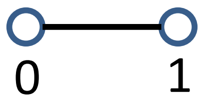


W_6

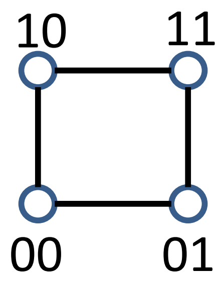
Some Special Simple Graphs

Definition: An n-cube, denoted by Q_n, is a graph that consists of 2^n vertices, each representing a distinct n-bit string. An edge exists between two vertices ⇔ the corresponding strings differ in exactly one bit position.

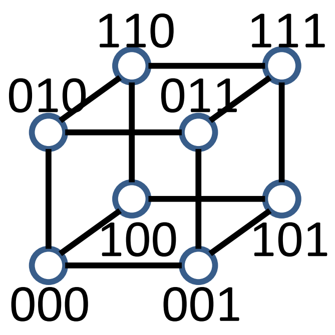
Examples :



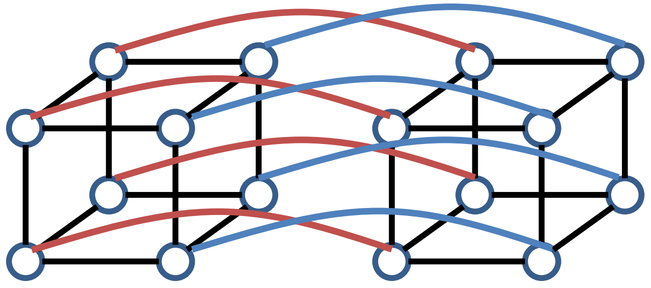
Q_1



Q_2



Q_3

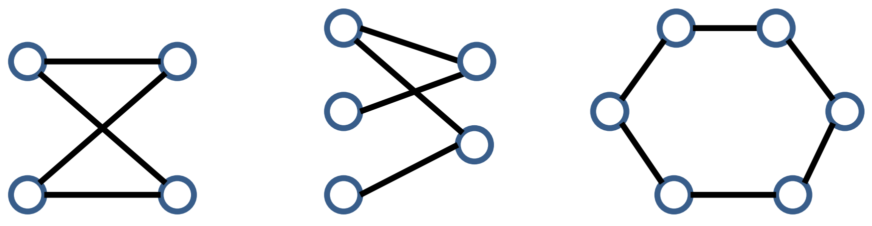


Q_4

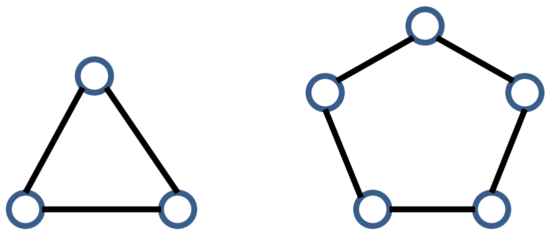
Some Special Simple Graphs

Definition: A bipartite graph is a graph such that the vertices can be partitioned into two sets V and W, so that each edge has exactly one endpoint from V, and one endpoint from W

Examples :



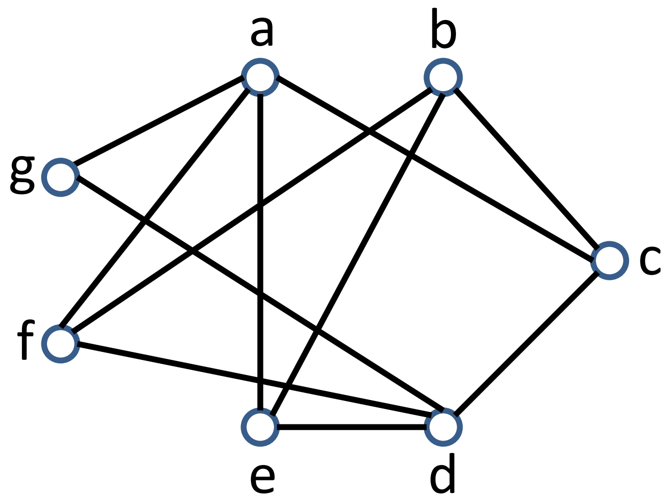
bipartite graphs

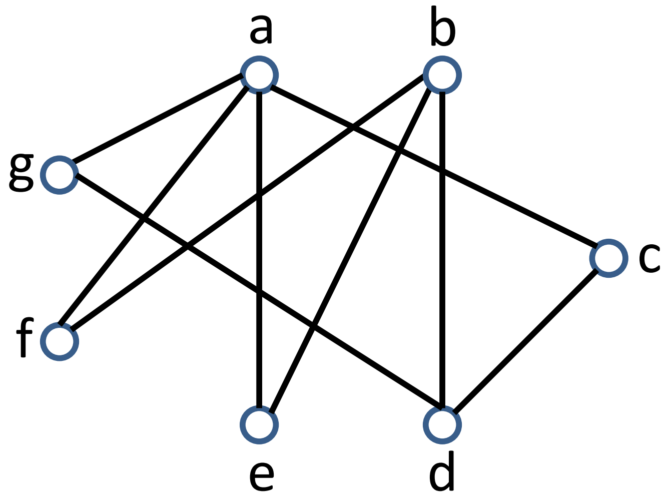


non-bipartite graphs

Some Special Simple Graphs

- Which of the following is a bipartite graph?





Check if a Graph is Bipartite

- The following is a very useful theorem :

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

- Proof : If there is a way to color the vertices, the same way shows a possible partition of vertices. Conversely, if there is a way to partition the vertices, the same way gives a possible coloring.

Check if a Graph is Bipartite

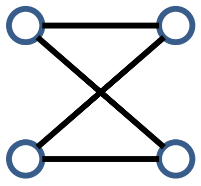
- The above implies the following algorithm to check if a connected graph is bipartite :

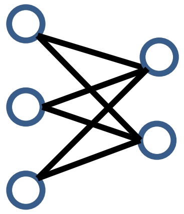
```
Step 1 :  Pick a vertex u.  Color it with white ;
Step 2 :  While there are uncolored vertices
              (i)   for each neighbor of a white vertex,
                        color it with black ;
              (ii)  for each neighbor of a black vertex,
                        color it with white ;
Step 3 :   Report YES if each edge is colored properly.
           Else, report NO ;
```

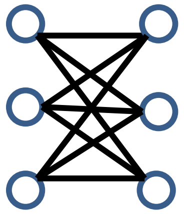
Some Special Simple Graphs

Definition: A complete bipartite graph $K_{m,n}$ is a bipartite graph with vertices partitioned into two subsets V and W of size m and n, respectively, such that there is an edge between each vertex in V and each vertex in W

Examples :


$K_{2,2}$

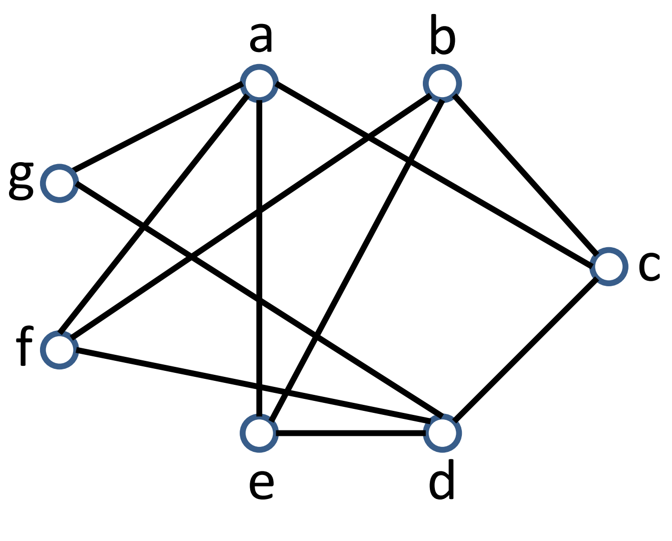

$K_{3,2}$

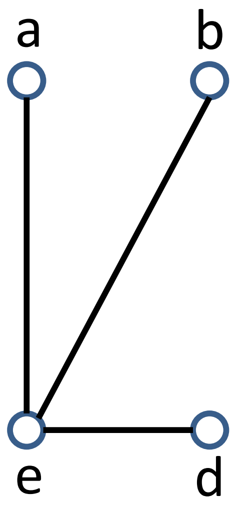

$K_{3,3}$

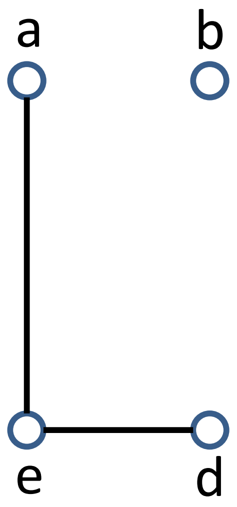
Subgraphs and Complements

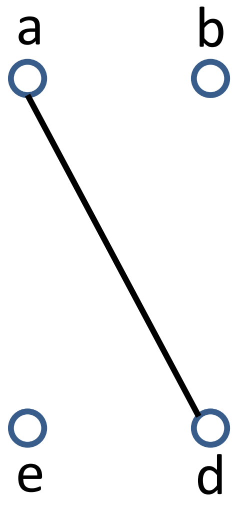
If G = (V, E) is a graph, then G' = (V', E') is called a **subgraph** of G if $V' \subseteq V$ and $E' \subseteq E$.

- Which one is a subgraph of the leftmost graph G ?







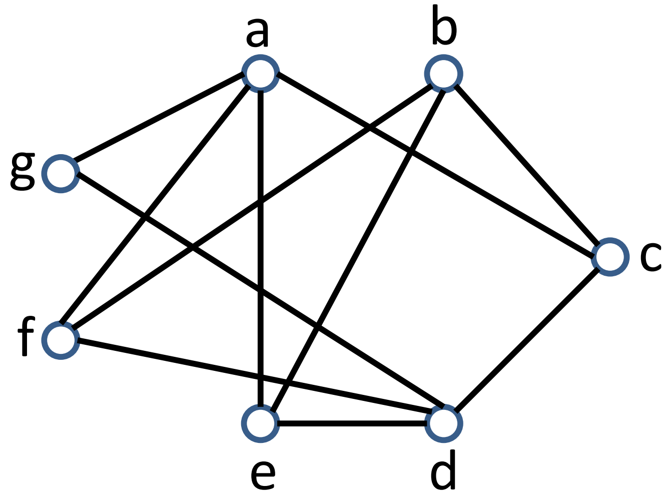


Subgraphs and Complements

If G = (V, E) is a graph, then the subgraph of G induced by U $\subseteq$ V is a graph with the vertex set U and contains exactly those edges from G with both endpoints from U

Ex : Consider the graph on the right side

What is its subgraph induced by the vertex set { a, b, c, g } ?



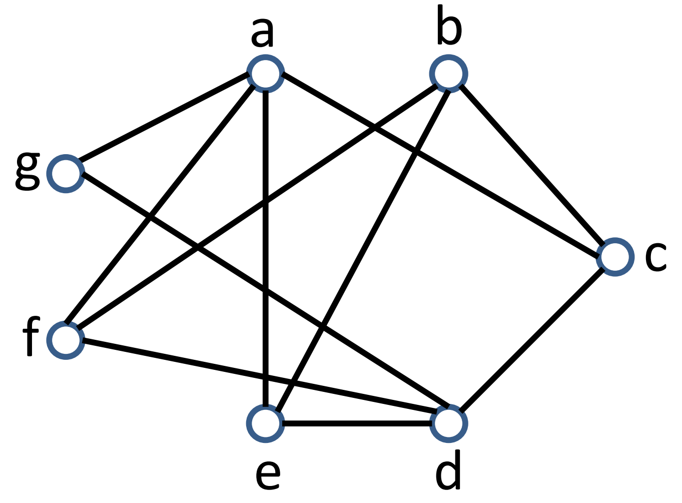
Subgraphs and Complements

If G = (V, E) is a graph, then the complement of G, denoted by $\overline{G}$, is a graph with the same vertex set, such that

an edge e exists in $\overline{G}$ ⇔ e does not exist in G

Ex : Consider the graph on the right side

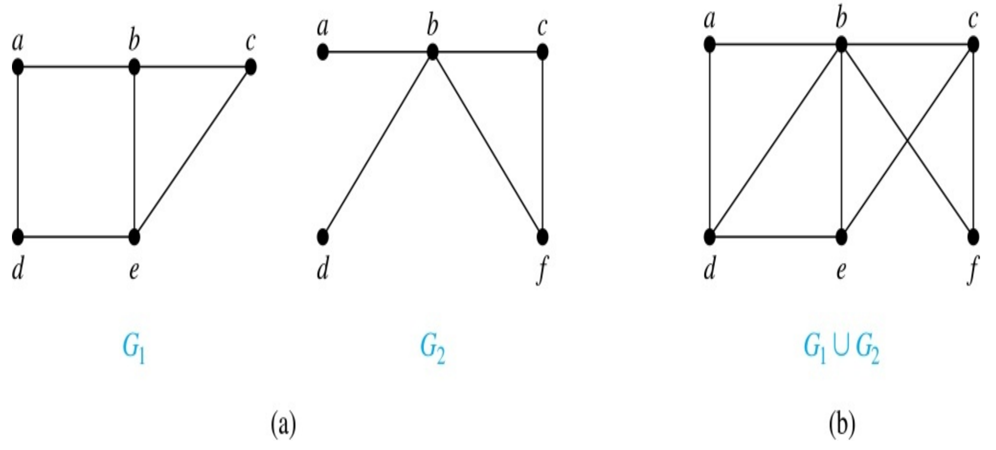
What is its complement ?



New Graphs from Old

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

Example:



Representing Graphs: Adjacency Lists

Definition: An *adjacency list* represents a graph (with no multiple edges) by specifying the vertices that are adjacent to each vertex.

Example:

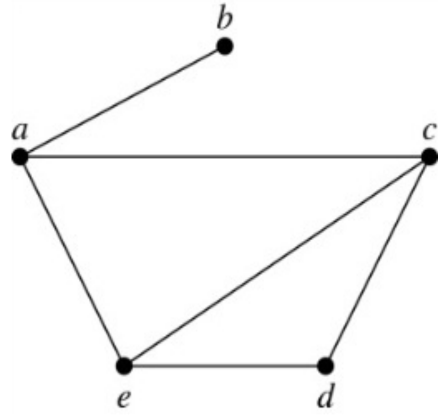


TABLE 1 An Adjacency List for a Simple Graph.

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

Example:

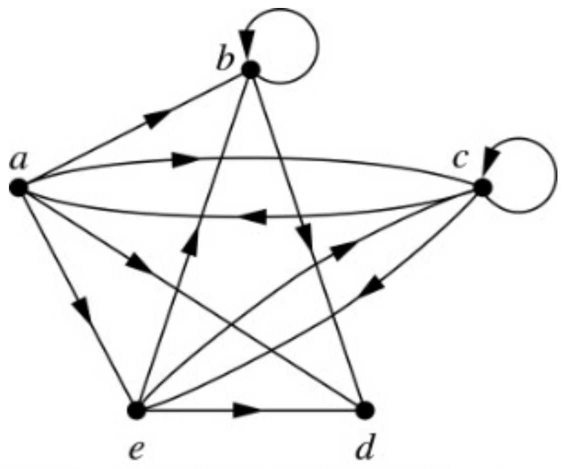


TABLE 2 An Adjacency List for a Directed Graph.

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

Representation of Graphs: Adjacency Matrices

Definition: Suppose that $G = (V, E)$ is a simple graph where $|V| = n$. Arbitrarily list the vertices of G as $v_1, v_2, \ldots, v_n$.

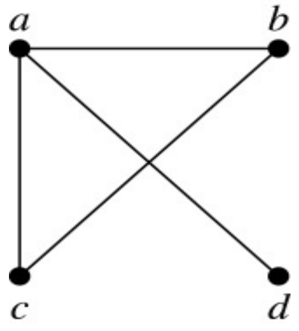
The *adjacency matrix*, **A**, of G, with respect to this listing of vertices, is the $n \times n$ 0-1 matrix with its (i, j)th entry = 1 when v_i and v_j are adjacent, and =0 when they are not adjacent.

- In other words: $A = [a_{ij}]$ and:

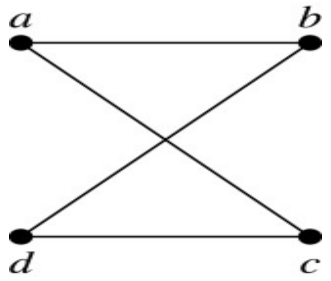
$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \text{ is an edge of } G, \\ 0 & \text{otherwise.} \end{cases}$$

Adjacency Matrices (*continued*)

Example: *The vertex ordering is is a, b, c, d.*



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



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

A sparse graph has few edges relative to the number of possible edges. Sparse graphs are more efficient to represent using an adjacency list than an adjacency matrix. But for a dense graph, an adjacency matrix is often preferable.

Note: The adjacency matrix of an undirected graph is symmetric:

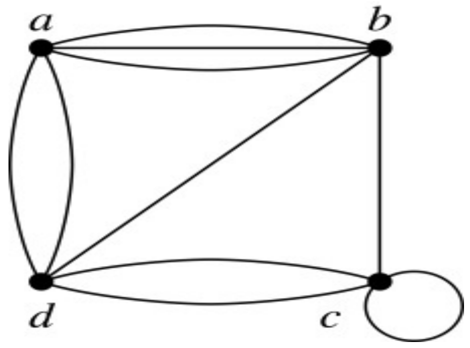
$$a_{ij} = a_{ji}, \forall i, j$$

Also, since there are no loops, each diagonal entry *is zero:* $a_{ii} = 0, \forall i$

Adjacency Matrices (*continued*)

- Adjacency matrices can also be used to represent graphs with loops and multi-edges.
- When multiple edges connect vertices *vi* and *vj*, (or if multiple loops present at the same vertex), the (*i*, *j*)th entry equals the number of edges connecting the pair of vertices.

Example: Adjacency matrix of a pseudograph, using vertex ordering *a*, *b*, *c*, *d:*



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

Adjacency Matrices (*continued*)

- Adjacency matrices can represent directed graphs in exactly the same way. The matrix A for a directed graph *G* = (*V*, *E*) has a 1 in its (*i*, *j*)th position if there is an edge from *vi* to *vj*, where *v*1, *v*2, … *vn* is a list of the vertices.
  - In other words,

$$a_{ij}=1 \quad if \quad (i,j)\in E$$
$$a_{ij}=0 \quad if \quad (i,j)\notin E$$

- Note: the adjacency matrix for a directed graph need not be symmetric.

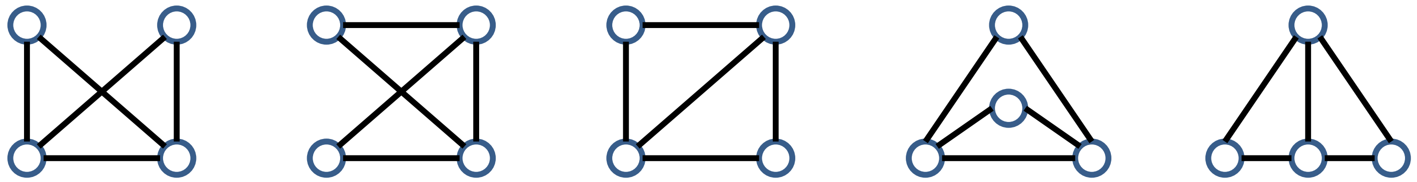
Graph Isomorphism

Graphs G = (V, E) and H = (U, F) are isomorphic if we can set up a bijection f : V → U such that

x and y are adjacent in G

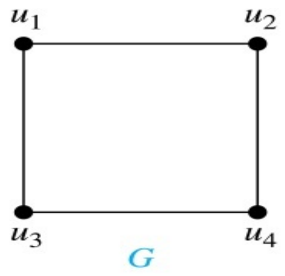
⇔ f(x) and f(y) are adjacent in H

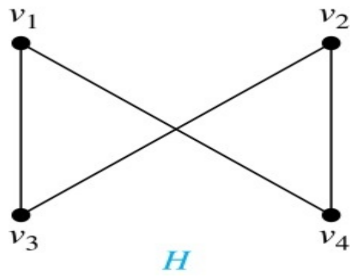
Ex : The following are isomorphic to each other :



Isomorphism of Graphs (*cont.*)

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





Solution: 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.

Isomorphism of Graphs (*cont.*)

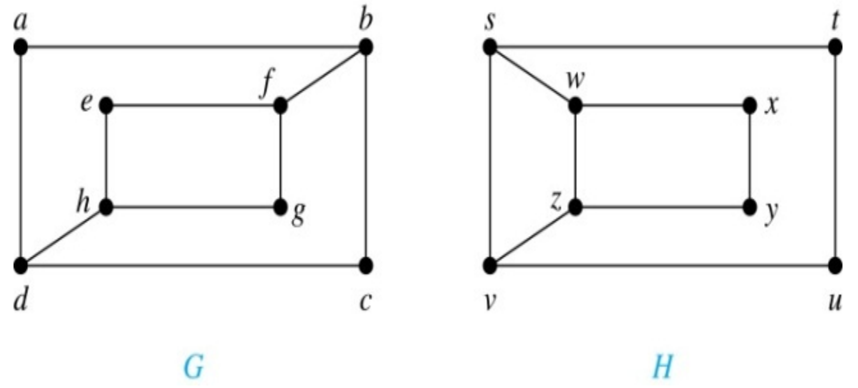
It is difficult to determine whether two graphs are isomorphic by brute force: there are $n!$ bijections between vertices of two n-vertex graphs.

Often, we can show two graphs are not isomorphic by finding a property that only one of the two graphs has. Such a property is called *graph invariant*:

- e.g., number of vertices of given degree, the degree sequence (list of the degrees), ...

Isomorphism of Graphs (*cont.*)

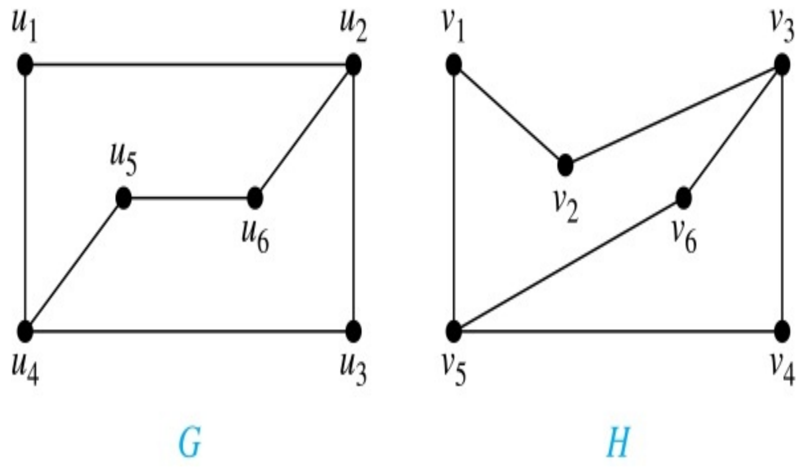
Example: Are these graphs are isomorphic?



Solution: No! Since *deg*(*a*) = 2 in *G*, *a* must correspond to *t*, *u*, *x*, or *y*, since these are the vertices of degree 2 in H. But each of these vertices is adjacent to another vertex of degree 2 in *H*, which is not true for *a* in *G*. So, G and H can not be isomorphic.

Isomorphism of Graphs (*cont.*)

Example: Determine whether these two graphs are isomorphic.



Solution: The function *f* is defined by: *f*(*u*1) = *v*6, *f*(*u*2) = *v*3, *f*(*u*3) = *v*4, *f*(*u*4) = *v*5 , *f*(*u*5) = *v*1, and *f*(*u*6) = *v*2 is a bijection.

Algorithms for Graph Isomorphism

- The best algorithms known for determining whether two graphs are isomorphic have exponential worst-case time complexity (in the number of vertices of the graphs).
- However, there are algorithms with good time complexity in many practical cases.
- See, e.g., a publicly available software called NAUTY for graph isomorphism.

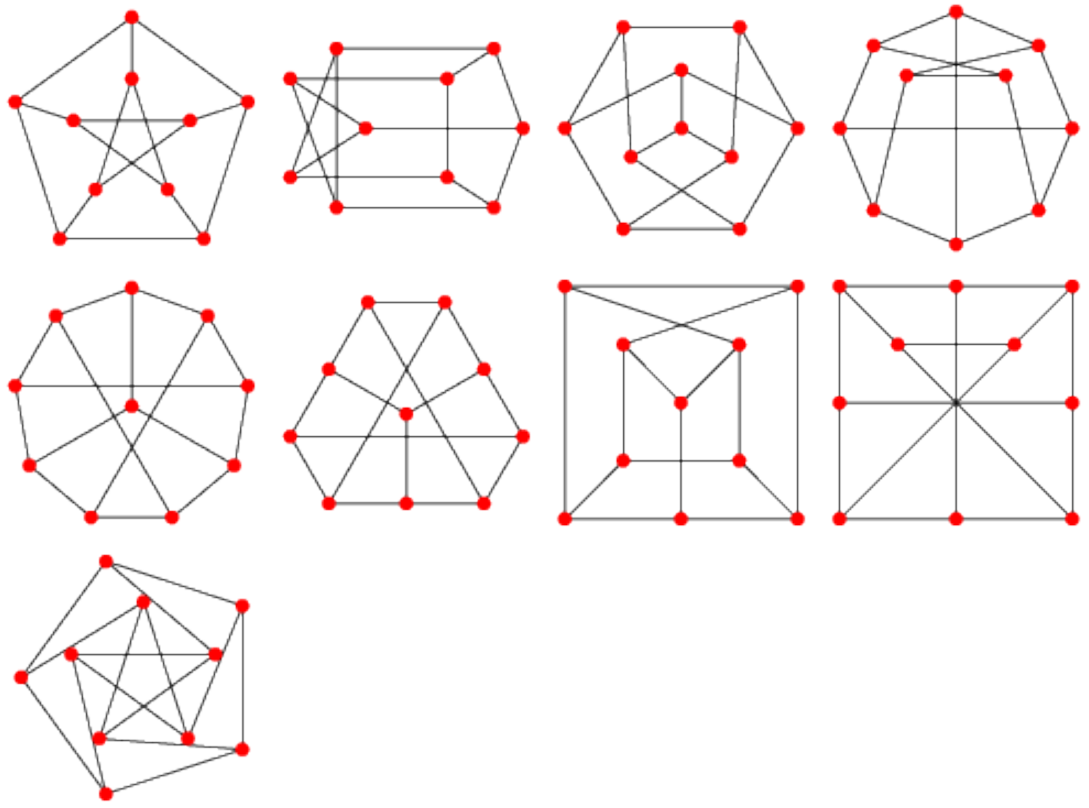
Applications of Graph Isomorphism

The question whether graphs are isomorphic plays an important role in applications of graph theory. For example:

Chemists use molecular graphs to model chemical compounds. Vertices represent atoms and edges represent chemical bonds. When a new compound is synthesized, a database of molecular graphs is checked to determine whether the new compound is isomorphic to the graph of an already known one.

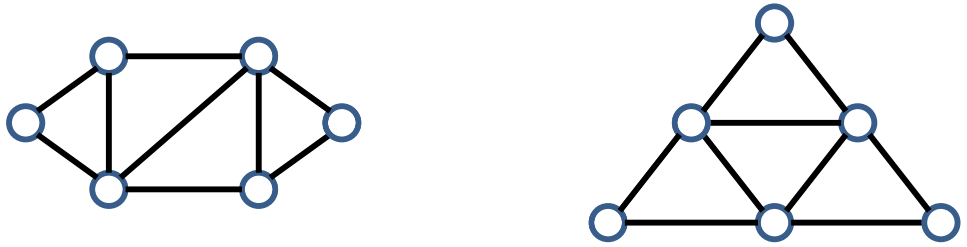
Graph Isomorphism

The following graphs are isomorphic to each other. This graph is known as the Petersen graph :



Graph Isomorphism

How to show the following are not isomorphic ?



How to show the following are not isomorphic ?