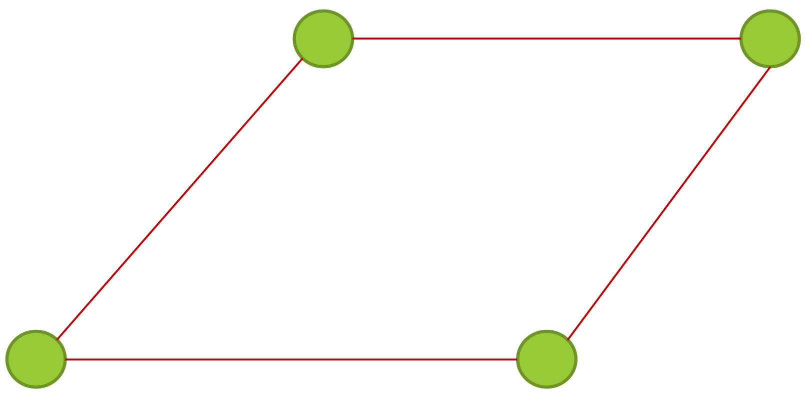
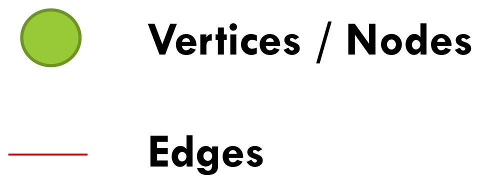
INTRODUCTION TO GRAPHS

A graph consists of vertices (nodes) and edges (connections between nodes). Graphs can represent various structures such as networks, relationships, and pathways.





REPRESENTATION OF GRAPHS

- **Adjacency List**
- **Adjacency Matrices**
- **Incidence Matrices**

1 **Adjacency List :** Adjacency lists represent a graph by listing adjacent vertices for each vertex.

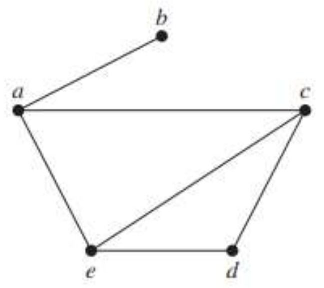


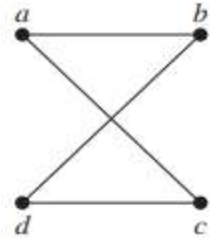
FIGURE 1 A Simple Graph.

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

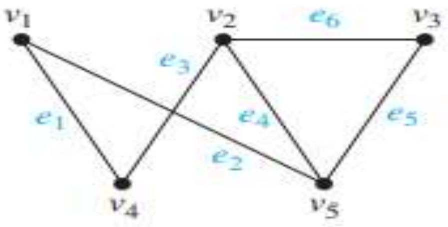
REPRESENTATION OF GRAPHS

2. **Adjacency Matrices :** Adjacency matrices represent a graph using a 2D matrix. Each entry represents whether an edge exists between two vertices.



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

3. **Incidence Matrices :** Incidence matrices represent a graph where edges are incident with vertices.



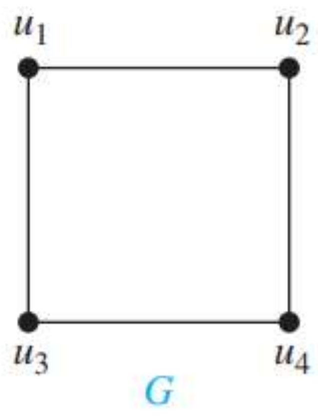
$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}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

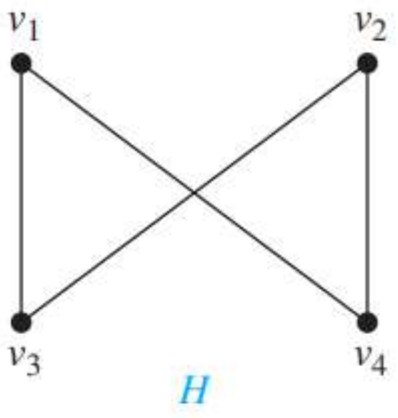
ADJACENCY LIST VS ADJACENCY MATRIX

Feature	Adjacency List	Adjacency Matrices
Storage	Uses less storage	Uses more storage
Best for	Simple sparse graph	Simple dense graph
Edge lookup	Slower	Faster
Edge insertion/deletion	Easy	Slightly hard
Ease of use	Easy to modify and manage	Easy for fast checks and matrix operations

ISOMORPHISM OF GRAPHS

Definition: If there is a one-to-one correspondence between vertices of the two graphs that preserves the adjacency relationship is called Isomorphism of graph.





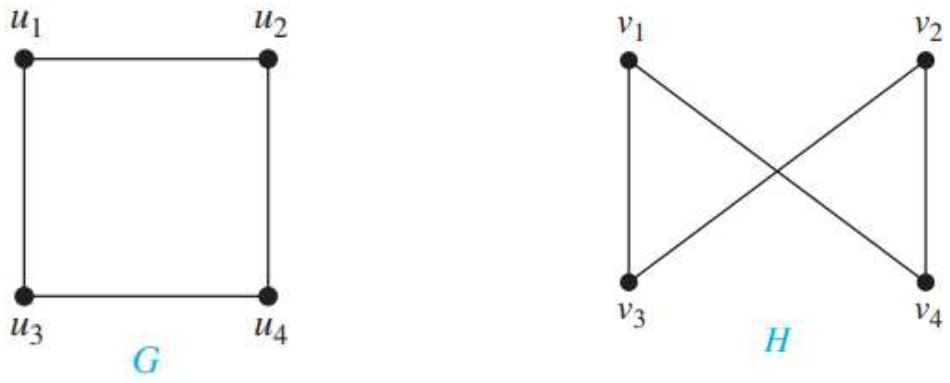
Graph G and H are Isomorphic

CONDITIONS OF ISOMORPHISM

Any two graphs will be known as isomorphism if they satisfy the following 4 conditions:

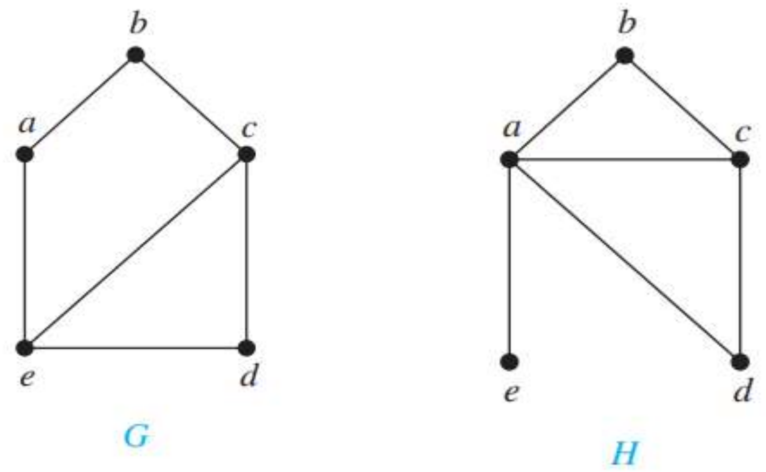
1. Equal number of **vertices** in the given graphs.
2. Equal number of **edges** in the given graphs.
3. Equal amount of **degree sequence** in the given graphs.
4. **One-to-one correspondence** between vertices of the two graphs that preserves the **adjacency relationship** between edges.

EXAMPLE OF ISOMORPHIC GRAPH



- They both have 4 number of vertices.
- They both have 4 number of edges.
- They both have equal amount of degree sequence (2,2,2,2).
- The function f with $f(u_1) = v_1$, $f(u_2) = v_4$, $f(u_3) = v_3$, and $f(u_4) = v_2$ is a one-to-one correspondence and they preserve adjacency relationship.

EXAMPLE OF NON-ISOMORPHIC GRAPH



- They both have 5 number of vertices.
- They both have 6 number of edges.
- Degree sequence of G (3,3,2,2,2) and degree sequence of H (3,3,2,2,1) are not equal.

ALGORITHMS FOR GRAPH ISOMORPHISM

- **Best Algorithm:** Depends on the numbers of vertices of graphs.
- **Linear Average-case Algorithms:** Some algorithms work with linear average-case time complexity. Generally it works faster than other algorithms.
- **NAUTY Software:** A practical tool that quickly tests graph isomorphism, handling graphs with up to 100 vertices in less than a second.
- **Special Algorithms:** Faster methods exist for certain types of graphs (like graphs with fewer connections).

APPLICATIONS OF GRAPH ISOMORPHISM

- **Chemistry:** Used to check if two chemical compounds with the same formula are structured differently by comparing their molecular graphs.
- **Electronic circuits:** Helps verify if the design of a circuit matches the layout created by automated tools.
- **Intellectual property in chips:** Used to detect if one company's chip design has copied parts from another company by finding matching graph sections.
- **Other fields:** Also used in bioinformatics and computer vision.