UNIT–1.0 (7 Hours) – Introduction to Graphs

1. What is a Graph

A **graph** is a mathematical structure used to represent relationships between objects. It is defined as **G = (V, E)** where:

- **V** → set of vertices (nodes)
- **E** → set of edges (connections)

Example: Social networks, computer networks, road maps.

---

2. Applications of Graphs

Graphs are widely used in:

- Computer networks (routing)
- Operating systems (deadlock detection)
- Social media (friend connections)
- Web page ranking (Google PageRank)
- Transportation networks
- Circuit design

---

3. Finite and Infinite Graphs

- **Finite Graph**: Limited number of vertices and edges
  Example: Graph with 5 vertices
- **Infinite Graph**: Infinite vertices or edges
  Example: Number line graph

---

4. Incidence and Degree

- **Incidence**: An edge is incident on a vertex if it connects to it.
- **Degree of a Vertex**: Number of edges incident on it
    - Loop contributes **2** to degree

---

5. Isolated Vertex, Pendant Vertex, Null Graph

- **Isolated Vertex**: Degree = 0
- **Pendant Vertex**: Degree = 1
- **Null Graph**: Graph with vertices but **no edges**

---

6. Walks, Paths, and Circuits

- **Walk**: Sequence of vertices and edges (repetition allowed)
- **Path**: Walk with no repeated vertices
- **Circuit (Cycle)**: Closed path (start = end)

---

7. Connected and Disconnected Graphs

- **Connected Graph**: Path exists between every pair of vertices
- **Disconnected Graph**: At least one pair has no path
- **Components**: Maximal connected subgraphs

---

8. Isomorphism of Graphs

Two graphs are **isomorphic** if:

- Same number of vertices
- Same number of edges
- Same adjacency structure

Used to identify structurally identical graphs.

---

9. Subgraphs

A **subgraph** is formed by selecting:

- A subset of vertices
- A subset of edges from the original graph

---

10. Euler Graphs

A graph is **Eulerian** if:

- It is connected
- All vertices have **even degree**

Euler Path: Uses every edge exactly once
Euler Circuit: Euler path that starts and ends at the same vertex

---

11. Operations on Graphs

Includes:

- Union
- Intersection
- Complement
- Deletion and contraction

---

12. Hamiltonian Paths and Circuits

- **Hamiltonian Path**: Visits every vertex exactly once
- **Hamiltonian Circuit**: Closed Hamiltonian path

---

13. Traveling Salesman Problem (TSP)

Find the **shortest Hamiltonian circuit** visiting each city once.

- NP-Hard problem
- Applications in logistics and routing

UNIT–2.0 (7 Hours) – Trees and Spanning Trees

1. Trees

A **tree** is a connected graph with **no cycles**.

Properties:

- If n vertices → n−1 edges
- Exactly one path between any two vertices

---

2. Pendant Vertices in a Tree

Vertices with **degree 1** are pendant (leaf nodes).

---

3. Distance and Center of a Tree

- **Distance**: Number of edges between vertices
- **Eccentricity**: Maximum distance from a vertex
- **Center**: Vertex with minimum eccentricity

---

4. Rooted and Binary Trees

- **Rooted Tree**: One vertex is designated as root
- **Binary Tree**: Each node has at most two children

Used in data structures and algorithms.

---

5. Counting Trees

Number of labeled trees with n vertices:
Cayley's Formula

$$n^{n-2}$$

---

6. Spanning Trees

A **spanning tree** includes:

- All vertices
- Minimum edges (n−1)
- No cycles

---

7. Fundamental Circuits

Adding one non-tree edge to a spanning tree forms a **fundamental circuit**.

---

8. Spanning Trees in Weighted Graphs

Find minimum cost spanning tree using:

- **Kruskal's Algorithm**
- **Prim's Algorithm**

UNIT–3.0 (7 Hours) – Cut Sets and Connectivity

1. Cut Set

A **cut set** is a set of edges whose removal disconnects the graph.

---

2. Cut Vertices

A vertex whose removal increases number of components.

---

3. Fundamental Cut Sets

Associated with spanning trees.

---

4. Connectivity and Separability

- **Connected graph** → No separation
- **Separable graph** → Has cut vertex

---

5. Network Flows

Deals with:

- Flow capacity
- Maximum flow (Ford–Fulkerson algorithm)

Used in traffic and data networks.

---

6. 1-Isomorphism and 2-Isomorphism

- **1-Isomorphism**: Based on cut vertices
- **2-Isomorphism**: Based on edge connectivity

UNIT–4.0 (7 Hours) – Planar and Dual Graphs

1. Planar Graph

A graph that can be drawn without edge crossings.

---

2. Kuratowski's Theorem

A graph is non-planar if it contains:

- $\mathbf{K_5}$
- $\mathbf{K_{3,3}}$

---

3. Detection of Planarity

Using Euler's formula:

$$V - E + F = 2$$

---

4. Dual Graphs

- **Geometric Dual**: Faces → vertices
- **Combinatorial Dual**: Based on incidence

---

5. Thickness and Crossings

- **Thickness**: Minimum planar layers
- **Crossing Number**: Minimum edge crossings

UNIT–5.0 (7 Hours) – Matrix Representation of Graphs

1. Incidence Matrix

Rows → vertices
Columns → edges

---

2. Circuit Matrix

Represents cycles in graph.

---

3. Fundamental Circuit Matrix

Derived from spanning tree circuits.

---

4. Cut-Set Matrix

Represents minimal cut sets.

---

5. Path Matrix

Represents paths between vertices.

---

6. Adjacency Matrix

- Square matrix
- A[i][j] = 1 if edge exists

UNIT–6.0 (7 Hours) – Coloring and Covering

1. Graph Coloring

Assign colors so that adjacent vertices have different colors.

---

2. Chromatic Number

Minimum number of colors needed.

---

3. Chromatic Polynomial

Gives number of colorings as a function of colors.

---

4. Coverings

- Vertex covering
- Edge covering

---

5. Four Color Problem

Every planar graph can be colored using **at most 4 colors**.