Mathematical Foundations of Computer Science

This Lecture: Graph Theory - Graph Representations

Department of Mathematical and Computational Sciences
National Institute of Technology Karnataka, Surathkal

MA714 (Odd Semester [2025-26])

Introduction

Graph theory is a fundamental area of discrete mathematics with applications in computer science, engineering, and various fields of science. A graph is a collection of vertices (or nodes) and edges that connect pairs of vertices. To study graphs effectively, we need suitable representations of graphs that allow efficient storage and computation.

Graph Representations

Graphs can be represented in multiple ways depending on the context, efficiency, and storage requirements. The three most common representations are:

1. Adjacency Matrix
2. Incidence Matrix
3. Adjacency List

Adjacency Matrix

An adjacency matrix is a 2D array of size $V \times V$ where V is the number of vertices in the graph. The entry A_{ij} indicates whether there is an edge between vertex i and vertex j.

For an undirected graph,

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge between the vertices } i \text{ and } j \\ 0 & \text{otherwise} \end{cases}$$

For a directed graph,

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge from the vertex } i \text{ to vertex } j \\ 0 & \text{otherwise} \end{cases}$$

Advantages

1. Simple to implement.
2. Checking adjacency is $O(1)$.

Disadvantages

1. Requires $O(V^2)$ space.
2. Inefficient for sparse graphs.

Incidence Matrix

An incidence matrix is a $V \times E$ matrix where V is the number of vertices and E is the number of edges. The entry B_{ij} indicates whether vertex i is incident to edge j.

For an undirected graph,

$$B_{ij} = \begin{cases} 1 & \text{if the vertex } i \text{ is incident to the edge } j \\ 0 & \text{otherwise} \end{cases}$$

For a directed graph,

$$A_{ij} = \begin{cases} 1 & \text{if the vertex } i \text{ is source to the edge } j \\ 0 & \text{otherwise} \end{cases}$$

Advantages

Provides a direct relation between vertices and edges.

Disadvantages

Requires $O(V \times E)$ space, which can be large for dense graphs.

Adjacency List

An adjacency list represents a graph as an array of lists. The index of the array represents a vertex, and each element in the list represents the vertices adjacent to that vertex.

Advantages

1. Requires $O(V + E)$ space, efficient for sparse graphs.
2. Iterating over neighbors of a vertex is efficient.

Disadvantages

Checking for the presence of a specific edge takes $O(degree)$ time.

Comparison of Representations

Adjacency Matrix

1. Best for dense graphs.
2. Quick adjacency checks.

Incidence Matrix

Useful for theoretical purposes and when edge-vertex relationships are crucial.

Adjacency List

1. Best for sparse graphs.
2. Space efficient.

1. Adjacency Matrix – useful in network flows, dense networks.
2. Incidence Matrix – important in algebraic graph theory and optimization problems.
3. Adjacency List – used in graph traversal algorithms (BFS, DFS).

Graph representations play a key role in graph theory and computer science. Choosing the appropriate representation depends on the structure of the graph and the type of operations to be performed. Understanding these representations helps in designing efficient algorithms for graph-related problems.