Session - 14

Graph

CONTENTS

01

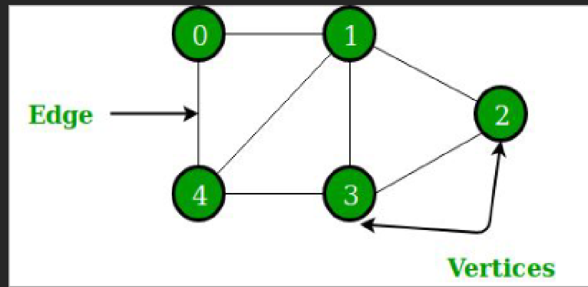
Introduction to Graph

Introduction to Graph

Definition:

- A **graph** is a collection of **nodes (vertices)** and **edges** that connect some pairs of nodes.
- Graphs model relationships between objects.

Trees vs Graphs:

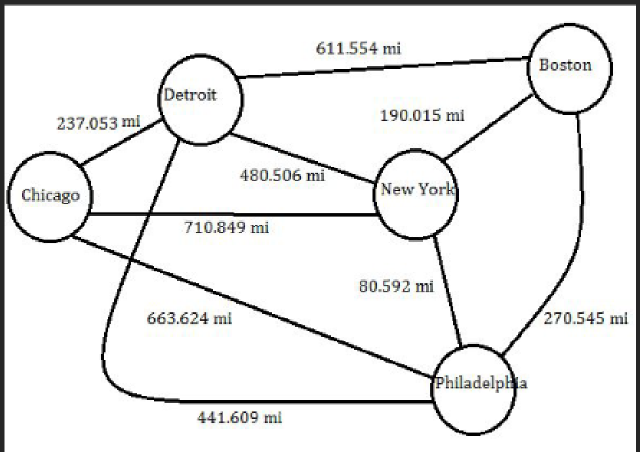


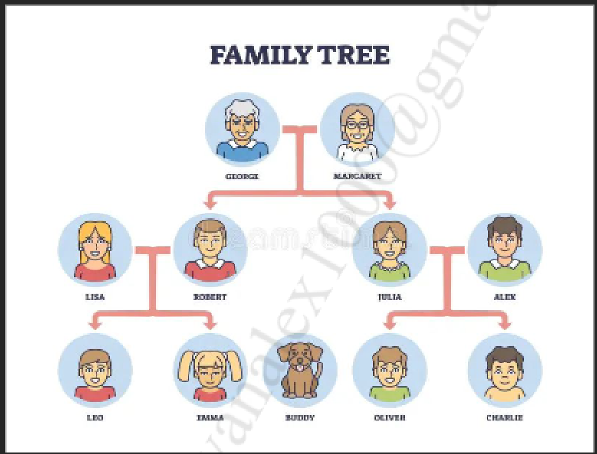
- All trees are graphs, but not all graphs are trees.
- Key differences:
  - **Trees** are always connected.
  - **Only one path** exists between any two nodes in a tree.
  - **No cycles** in a tree.
  - **Edges are undirected** in a basic tree.

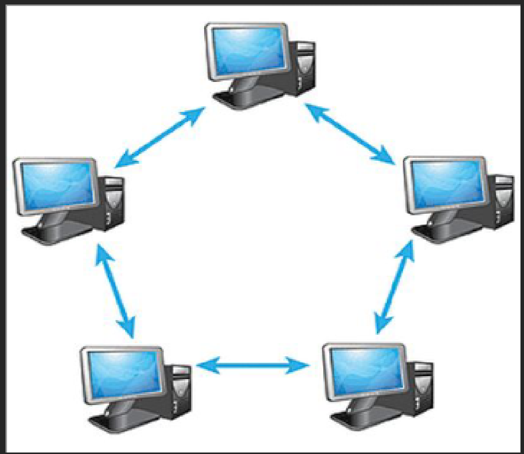
Introduction to Graph

Examples of where graphs are used:

- 🏠 **Houses** connected by **roads**
- 👨‍👩‍👧 **People** and their **family ties (Trees as well)**
- 💻 **Computers** in a **network**







02

Types of Graph

Types of Graph

Based on Direction:

- Directed Graph (DiGraph)
- Undirected Graph

Based on Weight:

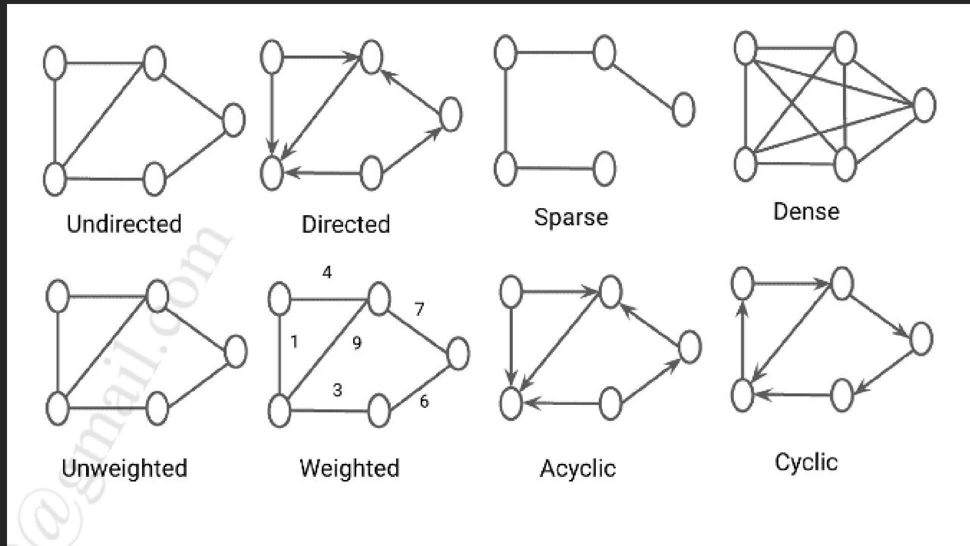
- Weighted Graph
- Unweighted Graph

Based on Cycles:

- Cyclic
- Acyclic (e.g., DAG - Directed Acyclic Graph)

Special Graphs:

- Complete, Connected, Tree, Bipartite



03

Graph Representation

Graph Representation

Edge List

- Shows list of edges in the graph as follows
  int[][] graph = {{1,2}, {2,3},{2,4},{3,4}}

- May not show all the nodes in the graph

Adjacency List

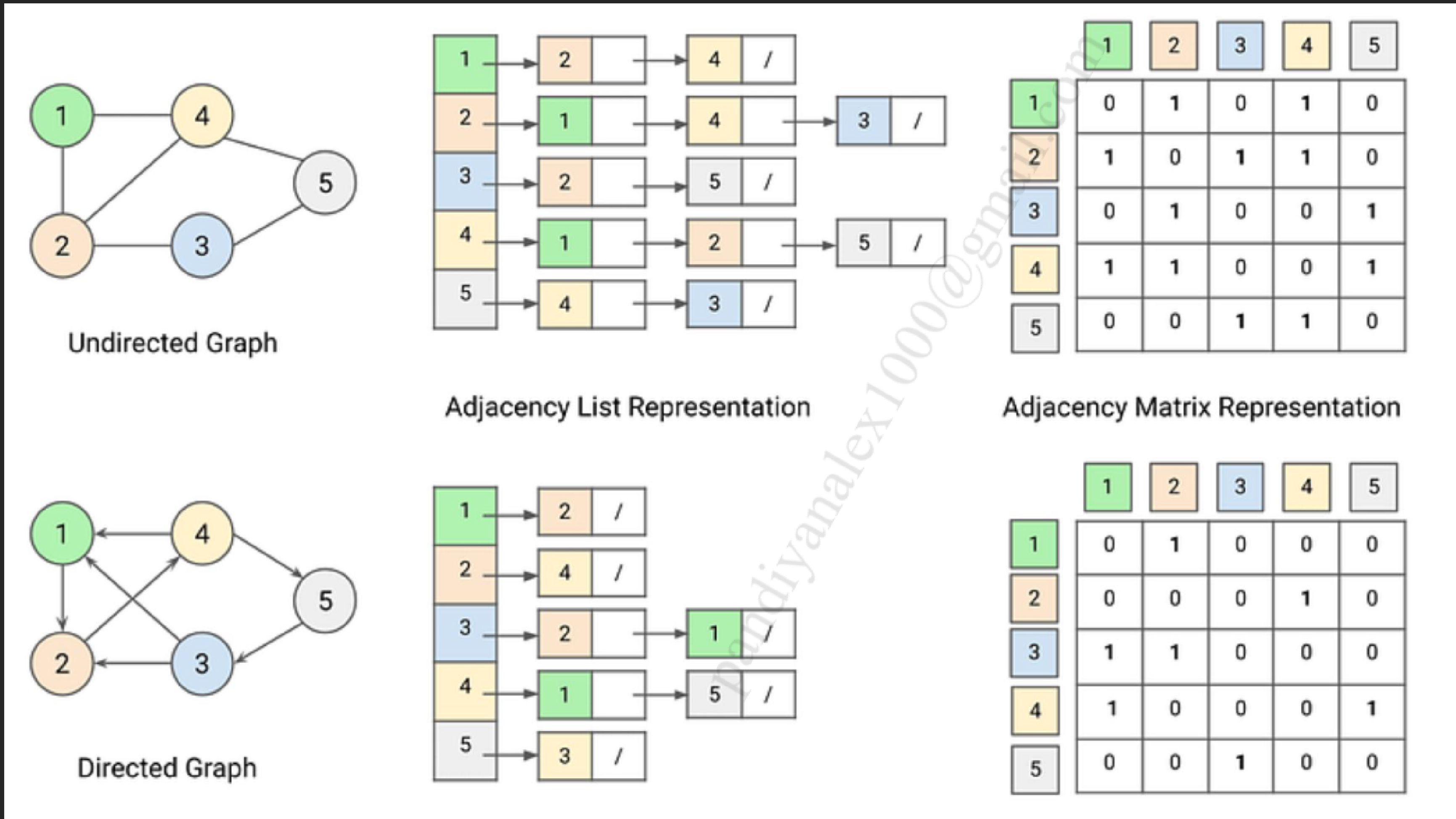
- Each node has entries with the entries representing the nodes it is connected to
  int[][] graph = {{2},{1, 3, 4},{2, 4},{2, 3}}

Adjacency Matrix

- Each row and column combination represents Vertices between 2 nodes. 1 indicates connection, 0 indicates no connection

```
int[][] graph = {{0, 1, 0, 0},
                 {1, 0, 1, 1},
                 {0, 1, 0, 1},
                 {0, 1, 1, 0}}
```

Graph Representation



04

Graph Traversal

Graph Traversal

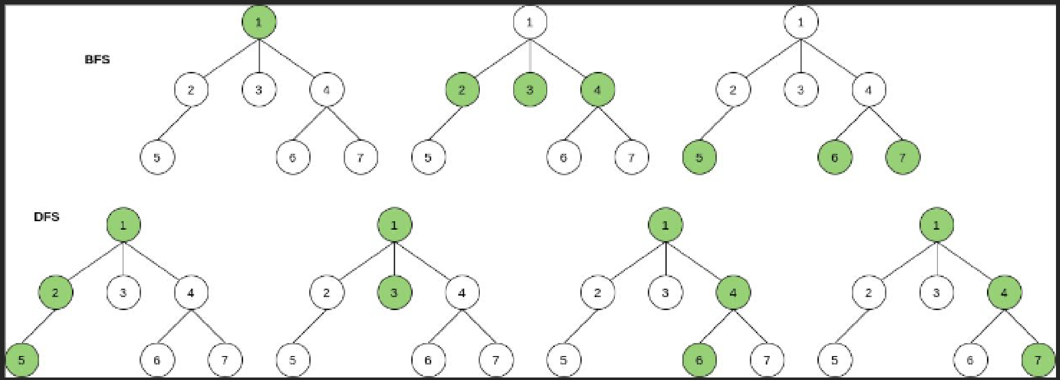
BFS - BFS strategy prioritizes the breadth over depth, it goes wider before going deeper

DFS - DFS prioritizes the depth over breadth

Comparison :-

- **BFS will find the shortest path** between the two points. DFS doesn't necessarily find the shortest path
- DFS on a (balanced) binary tree would take less memory than BFS
- **DFS is easier to implement** - recursively

The choice between BFS and DFS depends on the nature of the problem. Sometimes, both can be used



BFS vs DFS

Feature	BFS (Breadth-First Search)	DFS (Depth-First Search)
Technique	Level-order traversal	Recursive or backtracking traversal
Data Structure Used	**Queue**	**Stack** (explicit or call stack via recursion)
Backtracking	**No** backtracking – visits level by level	**Yes**, uses backtracking to explore all paths
Number of Edges Traversed	Up to all edges: O(V + E)	Up to all edges: O(V + E)
Optimality	**Yes**, finds the shortest path in unweighted graphs	**No**, may not find shortest path
Speed (Time Complexity)	O(V + E)	O(V + E)
Suitability (Decision Tree)	Great for finding the **shallowest solution**	Great for finding **any solution** or **deep solution**
Memory Efficient?	No — stores all neighbors at each level	Yes — only stores current path

05

Graph Applications & Algorithms

Graph Applications

1. GPS/Map routing (shortest path)
2. Social networks (friend suggestions)
3. Network packet routing
4. Web crawling (DFS/BFS)
5. Dependency resolution (DAG)

Graph Algorithms

1. **Dijkstra's Algorithm** – Shortest path from a source (weighted, no negative edges).
2. **Bellman-Ford Algorithm** – Shortest path with negative weights.
3. **Floyd-Warshall Algorithm** – All-pairs shortest paths.
4. **Prim's Algorithm** – Minimum spanning tree (greedy).
5. **Kruskal's Algorithm** – Minimum spanning tree (greedy + DSU).
6. **Kosaraju's Algorithm** – Another SCC detection method.
7. **Topological Sort** – Ordering DAGs.
8. **Union-Find (Disjoint Set Union - DSU)** – Used in Kruskal's, cycle detection.
9. ***A* Search Algorithm*** – Heuristic-based shortest path (used in games/AI).

06

Activity

Activity Graph

- Flood Fill
- 01 Matrix
- Clone Graph
- Number of Islands
- Rotting Oranges
- Word Ladder
- Word Search

Activity Graph

- Pacific Atlantic Water flow
- Cheapest Flights within K stops
- Topological Sort (DFS and Kahn's Algorithm)
- Course Schedule
- Alien Dictionary
- Connected Components in an Undirected Graph
- Number of Province

Activity Graph

- Accounts Merge
- Longest Increasing path in Matrix
- Course Schedule 2