Graph Basics – Comprehensive Summary

1. Definition of Graph
A Graph is a data structure used to represent relationships between different entities. It consists of a set of vertices (nodes) and a set of edges (connections). Mathematically, a graph is represented as G = (V, E), where V is the set of vertices and E is the set of edges.

2. Applications of Graph
Graphs are widely used in real-world systems such as social networks, computer networks, GPS navigation systems, database relationships, recommendation systems, and artificial intelligence algorithms.

3. Types of Graphs
Directed Graph: Edges have a specific direction ($u \rightarrow v$).
Undirected Graph: Edges have no direction ($u - v$).
Weighted Graph: Each edge has an associated weight (cost, distance, or time).
Unweighted Graph: Edges do not have weights.
Simple Graph: No loops or multiple edges are allowed.
Multigraph: Multiple edges between the same vertices are allowed.
Complete Graph: Every pair of vertices is connected by an edge.

4. Basic Terminology
Degree: Number of edges connected to a vertex.
In Directed Graphs: - In-degree: Number of incoming edges. - Out-degree: Number of outgoing edges.

Path: A sequence of vertices connected by edges. The length of a path is the number of edges.

Cycle: A path that starts and ends at the same vertex without repeating edges or vertices (except start/end).

Connected Graph: A path exists between every pair of vertices.
Disconnected Graph: The graph contains isolated components.

5. Loop and Multiple Edges
Loop: An edge that connects a vertex to itself. In undirected graphs, a loop contributes 2 to the degree.
Multiple Edges: More than one edge connecting the same pair of vertices.

6. Graph Representation
Adjacency Matrix: A 2D matrix where matrix[i][j] = 1 if there is an edge, otherwise 0. Advantages: Fast access. Disadvantages: High memory usage $O(n^2)$.

Adjacency List: Each vertex stores a list of adjacent vertices. Advantages: Memory efficient. Disadvantages: Slower edge lookup.

7. Counting Edges
In Undirected Graph: Total edges = (Sum of all degrees) / 2
In Directed Graph: Total edges = Sum of in-degrees = Sum of out-degrees

8. Graph Traversal
Breadth First Search (BFS): Uses a queue and explores vertices level by level. Used for shortest paths in unweighted graphs and connectivity checking.

Depth First Search (DFS): Uses recursion or stack and explores as deep as possible. Used for cycle detection, connected components, and topological sorting.

9. Graph vs Tree
Every tree is a graph, but not every graph is a tree. A tree must be connected, acyclic, and contain exactly (number of vertices – 1) edges.

10. Conclusion
Graphs are powerful data structures for modeling complex relationships. Choosing the correct representation and traversal algorithm is essential for efficiency.