1. Graph Traversal

Graph traversal is the process of visiting all the vertices of a graph in a systematic manner. The main purpose of graph traversal is to explore all nodes and edges of a graph. There are two common methods of graph traversal: Breadth First Search and Depth First Search. In Breadth First Search, vertices are visited level by level. In Depth First Search, traversal goes as deep as possible before backtracking. Graph traversal is important for searching, analyzing connectivity, and solving many graph-related problems.

2. Use of Queue and Stack in BFS and DFS

Queue and stack are used in BFS and DFS because they support the required order of traversal. Breadth First Search uses a queue to store vertices so that nodes are visited in the order they are discovered. This ensures level-by-level traversal. Depth First Search uses a stack to store vertices so that the most recently visited node is processed first. This helps DFS go deeper into the graph before exploring other paths.

3. Breadth First Search Example

Breadth First Search starts from a selected vertex and explores all its neighboring vertices first before moving to the next level. BFS visits vertices in a systematic order using a queue. This method ensures that the shortest path in terms of number of edges is found from the starting vertex.

```
Graph:
A - B - C
|   |
D   E

BFS Traversal starting from A:
A, B, D, C, E
```

4. Types of Network Flow Problems

Network flow problems deal with the flow of resources through a network. There are different types of network flow problems. The maximum flow problem focuses on finding the maximum possible flow from a source to a destination. The minimum cut problem focuses on finding the smallest capacity that separates the source and destination. Multi-source and multi-sink flow problems involve multiple starting and ending points. These problems are important in transportation, communication, and supply systems.

5. Applications of Topological Sort

Topological sorting is used in directed acyclic graphs. It is used in task scheduling where some tasks must be completed before others. It is used in project planning, course prerequisite scheduling, and compiler design. Topological sort helps in maintaining proper execution order and dependency management.

6. All Pair Shortest Path Problem

The all pair shortest path problem finds the shortest path between every pair of vertices in a graph. This problem is useful in networks where distances between all nodes are required. It helps in analyzing communication networks and transportation systems. Algorithms used for this problem consider all possible paths to determine minimum distances.

7. Bottleneck Capacity and Augmenting Path

Bottleneck capacity refers to the minimum capacity along a path in a flow network. It limits the maximum flow that can pass through that path. An augmenting path is a path from the source to the destination where additional flow can be pushed. These concepts are important in flow algorithms to increase total network flow.