GRAPHS-
Terminology and Representations of Graphs
Graph Implementation using STL
Graph Implementation in C++ without using STL
Implement Graph Data Structure in C
Graph Implementation in Java using Collections
Breadth First Search (BFS)   Iterative & Recursive Implementation
Depth First Search (DFS)   Iterative & Recursive Implementation
Arrival and Departure Time of Vertices in DFS
Types of edges involved in DFS and relation between them
Bipartite Graph
Determine if a given graph is Bipartite Graph using DFS
Minimum number of throws required to win Snake and Ladder game
Topological Sorting in a DAG
Kahn's Topological Sort Algorithm
Transitive Closure of a Graph
Check if an undirected graph contains cycle or not
Total paths in given digraph from given source to destination having exactly m edges
Determine if an undirected graph is a Tree (Acyclic Connected Graph)
2-Edge Connectivity in the graph
2-Vertex Connectivity in the graph
Check if given digraph is a DAG (Directed Acyclic Graph) or not
Disjoint-Set Data Structure (Union-Find Algorithm)
Chess Knight Problem – Find Shortest path from source to destination
Check if given Graph is Strongly Connected or not
Check if given Graph is Strongly Connected or not using one DFS Traversal

Union-Find Algorithm for Cycle Detection in undirected graph

Kruskal's Algorithm for finding Minimum Spanning Tree

Single-Source Shortest Paths – Dijkstra's Algorithm

Single-Source Shortest Paths – Bellman Ford Algorithm

All-Pairs Shortest Paths – Floyd Warshall Algorithm

Find Cost of Shortest Path in DAG using one pass of Bellman-Ford

Least Cost Path in Weighted Digraph using BFS

Find maximum cost path in graph from given source to destination

Determine negative-weight cycle in a graph

Print all k-colorable configurations of the graph (Vertex coloring of graph)

Print All Hamiltonian Path present in a graph

Greedy coloring of graph