

Graph

✓ Traversal

- * DFS
- * BFS

✓ Shortest Path Print / Distance

- * BFS
- * DFS
- * Diameter [tree]-
- * 0-1 BFS
- * Dials Algorithm
- * Print all path from a source to destination using DFS
- * Print all path using BFS

✓ Connected Component in graph

- * DFS
- * BFS
- * DSU

✓ Cycle detection undirected

- * BFS
- * DFS
- * DSU
- * Detect Negative Cycle of a Graph [Using bellman ford]
- * Print Cycle path using back-edge [which creates cycle]

✓ Cycle detection directed

- * BFS - Kahn's Algorithms
- * DFS, Stack concept
- * DSU
- * Negative Cycle exist
- * Print negative Cycle of graph [use bellman Ford algo]

✓ Topological Sort

- * BFS (Kahn's Algo)
- * DFS

✓ Bipartiate Graph

- * BFS
- * DFS
- * Graph / Edge colouring

✓ Single Source Shortest Path

- * Bellman Ford [(V-1)*edge distance]
- * Dijkstra algorithm

✓ **Multisource Shortest Path**

- * Floyd Warshall [all pair distance] [Src - inter - dest]

✓ **DSU**

- * Union By Rank
- * Union By Size

✓ **Minimum Spanning Tree**

- * Prims /Jarnik's Algorithm
- * Kruskal's Algorithm[DSU]

✓ **Strongly Connected Components**

- * Kosaraju's Algorithm[DFS]
- * Tarjan's algorithms - conceptual
- * Cycle length and Node

✓ **Bridges (Cut edge)** [$low[x] > dis[x]$]

- * Tarjan's algorithms
- [For better Experience Saradha didi Graph]

✓ **Articulation Point (Cut vertices)** [Advance] [$low[x] \geq dis[x]$]

- * Discovery time
- * Lowest time
- * Tarjan's algorithms
- * Back-edges

✓ **Re-Rooting of Trees**

- * Problem discuss[brute-forces]
- * Re-rooting

✓ **Maximum Network Flows** [most Advance topic-not for newbie]

- * Ford-Fulkerson/Edmonds-Karp
- * Dinic Algorithm
- * Residual Networks and Augments path