Graph Algo:-

→ Introduction:- Graph → relationship between Objects
→ combo of Nodes and Edges

Real life Application:-

⇒ Google Maps → to find route (shortest Time or Distance)

⇒ Social Network ⇒
User == Vertex
Connection == Edge

⇒ Web Search ⇒ (pages on internet are linked by hyperlinks)
- each page == vertex
- link between 2 page == Edge

Types:-
- Directed
- Undirected
- Cyclic
- Weighted

Graph representation

Adjacency matrix

⇒ Space Complexity $O(V^2)$

V = No. of Vertex

⇒ Time to access → $O(1)$

Adjacency List

Space Complexity $O(V+E)$

Vertex Edge

⇒ Graph Traversal :- Visiting every vertex & Edge once in a well-defined order

ⓐ BFS (Breadth First Search

Time Complexity → $O(V+E)$ & Space → $O(V)$

Application of BFS :-

→ shortest path & Minimum Spanning tree for unweighted graph

→ P2P Network → Crawlers in Search Engine

Implement

→ Cycle Detection in Undirected graph → Path finding

→ GPS Navigation :- BFS is used to find all neighboring locations

→ Social Networking Websites :- In social n/w, we can find people with a given distance 'K' from a person using BFS till 'K' levels

(b) DFS (Depth first Search)
↳ recursive, uses backtracking

Time Complexity → O(V+E)

Application:- ① Cycle detection
② Path finding
③ Topological sorting
④ Find Strongly Connected Components

Implement all these

To Implement

⑧ for find no of Connected Components

© MST (Minimum Spanning Tree)

→ what is Spanning Tree
→ what is MST ?

Application of MST :-

① Cluster Analysis
② Handwriting Recognition
③ Image Segmentation

Famous Algo for MST :-

① Kruskal's Algo ↔ Greedy Algo
Time complexity → O(V log E)

② Prim's Algo → Greedy Algo
O(E log V) with (heap)

Shortest Path Algo:-

① Single Source Shortest Path Algo (SSSP)
(find shortest of single vertices to all other vertices)

(A) Dijkstra's Algo:- [Edge weight ↔ Non -ve]

Time Complexity → $O(V^2)$ but with heap it becomes, $O(V + E \log V)$

(B) Bellman ford Algo:- [No Negative weight cycles]

Time Complexity → $O(VE)$

↑ will not work for

All-pair Shortest Path:-

→ Floyd Warshal's Algo → $O(V^3)$

→ Johnson's Algo → $O(V^2 \log V + VE)$

Must Attempts

→ Implement BFS based Algo to find the shortest route in Snake & Ladder Game

→ Implement a program to classify each of the graph as "forward Edge", "Back Edge" or "Cross Edge"

→ Read About Hamiltonian Cycle

→ Implement "Travelling Salesman Problem" using DP

→ Learn about → Articulation Point
→ Bridge in Graph

→ Kosaraju's Algo & Tarjan's Algo

→ Solve "HOLI" on SPOJ

→ Implement "Flood Fill Algo"

→ Implement an algo for "Splitwise App"

it's me BABBAR :)