

Graph Algo:

→ Introduction: Graph → relationship between objects
→ combo of Nodes and Edges

Real life Application:-

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

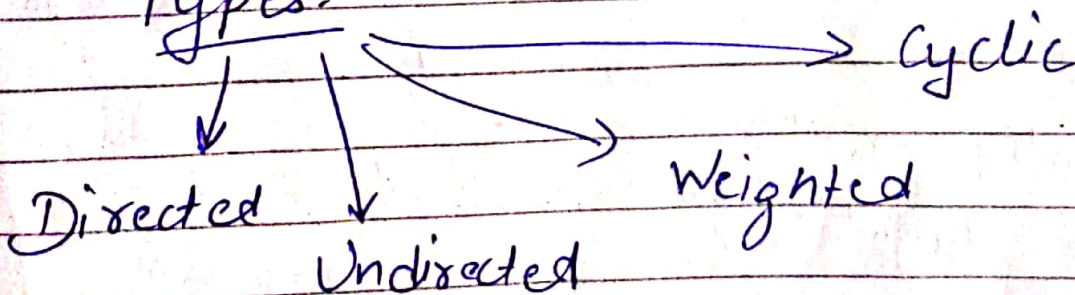
⇒ Social Network ⇒ User == Vertex

Connection == Edge

⇒ Web Search ⇒ (pages on internet are linked by hyperlinks)

- each page == vertex
- link between 2 page == Edge

Types:-



graph representation

Adjacency
matrix

Adjacency
List

→ Space Complexity $O(V^2)$

↓
 V^2 No. of
Vertex

Space Complexity $O(V + E)$

↓ ↓
Vertex Edge

→ Time to access $O(1)$

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

① BFS (Breadth First Search)

Time Complexity $\rightarrow O(V + E)$ & Space $\rightarrow O(V)$

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 ⇒ In social n/w, we can find people with a given distance 'K' from a person using BFS till 'K' levels

⑥ DFS (Depth first search)

↳ recursive, used 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

⑨ 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 \leftrightarrow Greedy Algo

Time Complexity $\rightarrow O(V \log E)$

② Prim's Algo \rightarrow Greedy Algo

$O(E \log V)$ with heap

Shortest Path Algo:-

① Single Source Shortest Path Algo (SSSP)

(find shortest of single vertex to all other vertices)

② Dijkstra's Algo:- [Edge weight \leftrightarrow Non -ve]

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

③ Bellman ford Algo:- [No Negative weight cycles]

Time $\Rightarrow O(VE)$
Complexity

I will not work for

All-pairs Shortest Path:

\rightarrow Floyd warshall's Algo $\rightarrow O(V^3)$

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

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 😊