**GRAPH THEORY**

TYPES OF GRAPHS:

1.Undirected Graphs: A undirected graph is a graph where edges have bidirectional, which means a (u, v) edge is equal to (v, u) edge.

2.Directed Graphs: A Directed graph is a graph where edges have directional, which means a (u, v) edge is not equal to (v, u) edge.

3.Weighted Graphs: A graph that contain weights on the edges are called as weighted Graphs.

SPECIAL GRAPHS:

4. Trees: Trees are the undirected and connected Graphs

With NO cycles, and with N Nodes and N-1 Edges.

5.Rooted Tree:is a tree with a designated root node where every edge either points away from or towars the root node. (In-tree and out-tree)

6.DAGs: Directed Acyclic Graphs are the directed Graphs with NO cycle.

7.Bipartite Graph: is one whose vertices can be split in to two independent groups U, V such that every edge connnects betweens U and V.

Ex: Two colorable graph or the is no odd length cycle.

8.Complete Graph: A Graph in which every edge connects to all other Edges.

REPRESENTATION OF GRAPHS:

1.Adjacency Matrix: A way to represent graph in a matrix of V\*V.

A= [0 1 1 0] 0--- No edge Between them

[0 0 0 1] 1--- Edge Between them

[1 1 1 1] Any number ---Weight on that edge

[1 0 0 0]

2.Adjacency List: Where Nodes maps to list of edges connected to it. (VIMP)

A--- B, C, D, E

B--- A, D

C--- A, F

D--- A, B

3.Edge List: (u, v, w) Edge from u to v with Weight w, unordered List of Edges.

PROBLEMS ON GRAPHS:

1.Shortest Path Problem: Shortest path from node A to node B.

ALGOs: BFS (Unweighted Graph)

Dijkstra’s Algo (Works only with positive)

Bellam-Ford (Works with negative edges)

Floyd-Warshall (Using DP)

2.Connectivity: Does Path exists between Node A to Node B.

Solution: Use Union Find D.S or search Algo (DFS)

3.Negative Cycles: Does Negative cycle exists in a graph.

ALGOs: Bellman-Ford and Floyd-Warshall

4.Strongly Connected Components: Can be thought of as Self-Contained Cycles withthin a Directed Graph where every vertex in a given cycle can reach every other vertex in the same cycle.

ALGOs: Tarjans’s and kosaraju’s Algorithm

5.Travelling salesman Problem: Given a list of cities and distances, what is the shortest possible route that visits each city exactly once and returns to the origin city.

NP-HARD problem: Very computationally challenging pro

Algo: Held-karp, Branch and Bound and many Approximation Algos.

6.Bridges: A bridge/Cut edge is any edge in a graph whose removal increases the number of connected Components.

7.Articulation Points: is any node in a graph whose removal increases the number of connected components.

8.MST: Minimum Spanning tree is a subset of the edges of a connected, edge-weighted graph. Mi nimum Possible total edge weight connects all the vertices.

ALGOs: Kruskal’s, Prim’s and Boruka’s Algorithm

9.Network Flow: Max Flow

With an Infinite input source how much “flow” can we push through the network.

ALGOs: Ford-Fulkerson, Edmonds-Karp and Dinic’s Algo

DEPTH FIRST SEARCH

Is the most fundamnetal search Algo used to explore nodes and edges of a graph. O (V+E)

Count Connected Components, determine connectivity, Find Bridges/A.Points then DFS Shines

Compute a MST’s iof Graph

Detect and find cycles in a graph

BREADTH FIRST SEARCH

Is the most fundamnetal search Algo used to explore nodes and edges of a graph. O (V+E)

Main Problem:Shortest Path on unweighted Graphs