CL-## Data Structures – Lab

Problem#01: Adjacency Matrix – Network Connectivity Test **(DFS/BFS)**

A company has a network of offices connected by communication lines. To ensure reliable communication, you need to verify if all offices are connected.

Task: Use adjacency matrix to store the no. of office's as vertices. Write a program to check if the network is fully connected using **DFS** or **BFS** traversal. If the network is not connected, determine the number of isolated office groups.

Number of offices: 6

Edges: [(1, 2), (2, 3), (4, 5)]

Sample Output: Network is **not fully** connected. Number of **isolated** groups: **3**

Problem#02: Adjacency List – BFS - Fastest Delivery Route

A delivery service wants to determine the shortest route between two warehouses in an unweighted road network. **Task**: Use adjacency list to store the graph vertices and write a program using **BFS** to find the shortest path between two warehouses. Output the **path length** and the **actual path.**

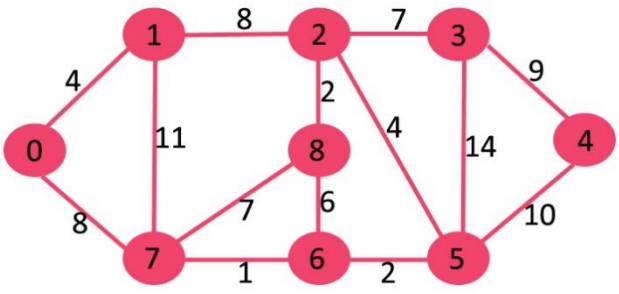
Input: Number of warehouses: 7
Roads/Edges: [(1, 2), (1, 3), (2, 4), (3, 5), (5, 6), (6, 4)]
Start: 1 End: 6
Output: Shortest route length: **3** and Route: 1 -> 3 -> 5 -> 6

Problem#03: Minimum Cable Cost for Internet Setup **(Kruskal's Algorithm)**

A company is setting up internet cables between office buildings. Each cable has a cost, and the company wants to minimize the total cost while ensuring all buildings are connected.

Task: Use **Kruskal's algorithm** to find the **minimum spanning tree (MST)** of the cable network. Output the total cost of the MST and the selected connections.

Input: Use the below weighted graph and find the MST?



Problem#04: Optimized Electricity Grid (**Prim's Algorithm**)

A city is building an electricity grid to connect all districts. The cost of installing power lines between districts is given. Find the cheapest way to connect all districts.

Task:
Use **Prim's algorithm** to calculate the **minimum spanning tree (MST)** starting from a given district. Output the total cost and the selected connections.

Input: Number of districts: **4**

Power line costs (u, v, w): [(1, 2, 1), (1, 3, 4), (2, 3, 2), (3, 4, 3), (4, 1, 5)] where **u,v** are the vertices and **w** is their respective weight/cost.

Start District: **1**

Output: Minimum cost: 7

Selected power lines: [(1, 2), (2, 3), (3, 4)]