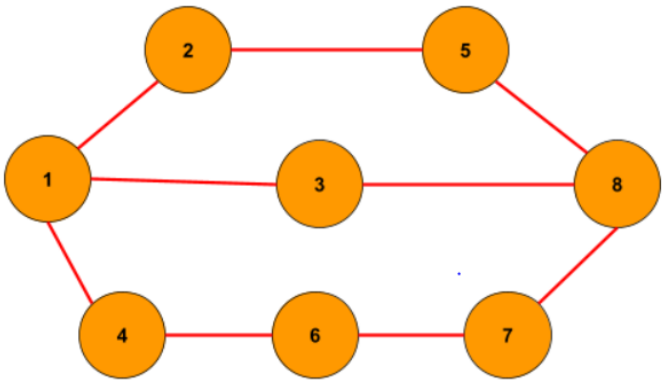
Find the Shortest Path in an Unweighted Graph

CodeStudio

Let **G = (V, E)** be an undirected graph with **E** edges and **V** vertices. Let **T** be the shortest path between any 2 vertices in the graph such that there is no other path in the graph between any 2 vertices whose sum of edge weights is less than **T'**s sum of edge weights.

In the case of unweighted graphs, there will be no edge weights. In that case, the shortest path **T** will become the path between the given 2 vertices with the minimum number of edges.

Example:



Source: 1, Destination: 8 Output: {1, 3, 8}

addEdge Function:

- **Purpose:**
  - Populates the graph's adjacency list based on the provided edge list.
- **Explanation:**
  - Iterates through each edge in the **edges** vector.
  - For each edge, extracts the source vertex **u** and iterates over the connected vertices.
  - Adds an edge from **u** to **v** in the adjacency list.
  - For undirected graphs, adds an edge from **v** to **u**.
- **Time Complexity:**
  - **O(E), where E is the number of edges in the input vector.**
- **Space Complexity:**

- **O(E), where E is the number of edges. Each edge results in the creation of an entry in the adjacency list.**

Approach 1: Function to find the shortest path from source to destination using BFS

- **Purpose:**
  - Finds the shortest path from a source node to a destination node in the graph using BFS.
- **Explanation:**
  - Initializes data structures such as **visited**, **parent**, and a **queue** for BFS traversal.
  - Performs BFS traversal starting from the source node.
  - Tracks visited nodes, parent nodes, and enqueues nodes for exploration.
  - Reconstructs the shortest path from destination to source using parent links.
- **Time Complexity:**
  - **O(V + E), where V is the number of vertices and E is the number of edges.**
  - **Combined with addEdge Function:**
    - **Total Time Complexity: O(V + E) + O(E) = O(V + 2E) ≈ O(V + E)**
- **Space Complexity:**
  - **O(V + E), where V is the number of vertices and E is the number of edges.**
  - **Combined with addEdge Function:**
    - **Total Space Complexity: O(V + E)**