DAILY PROGRAMMING

CHALLENGE

WISDOM SPROUTS

IT Training Hub

Find the Shortest Path in an Unweighted Graph

You are given an unweighted graph represented by an adjacency list. Your task is to find the shortest path (in terms of the number of edges) between two given nodes in the graph.

Input:

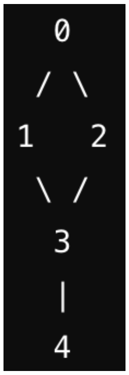
- An integer V representing the number of vertices in the graph.
- A list of edges, where each edge connects two vertices of the graph.
- Two integers, start and end, representing the source and destination nodes respectively.

Output:

- Return the shortest path length (number of edges) from start to end. If there is no path, return -1.

Examples:

- Example 1
  Input: V = 5
  Edges = [[0, 1], [0, 2], [1, 3], [2, 3], [3, 4]]
  start = 0
  end = 4
  Output: 3
  Explanation:



```
  0
 / \
1   2
 \ /
  3
  |
  4
```

The shortest path from node 0 to node 4 is through nodes 0 -> 2 -> 3 -> 4, which has 3 edges.

Constraints:

- $1 \leq V \leq 10^4$
- $0 \leq E \leq 10^4$
- The graph is undirected and can contain multiple edges between the same pair of nodes.
- There are no self-loops.

Test Cases:

1. Input: V = 5
   Edges = [[0, 1], [0, 2], [1, 3], [2, 3], [3, 4]]
   start = 0
   end = 4
   Output: 3
2. Input: V = 3
   Edges = [[0, 1], [1, 2]]
   start = 0
   end = 2
   Output: 2
3. Input: V = 4
   Edges = [[0, 1], [1, 2]]
   start = 2
   end = 3
   Output: -1

Edge Cases:

1. If start and end are the same and there are no edges, the path length is 0.
2. If there is no connection between start and end, return -1.
3. If the graph is fully connected, ensure the shortest path is calculated correctly.