Practice Problems on BFS and DFS

Problem 1: Graph Traversal

You are given a directed graph represented using adjacency lists. Implement both Breadth-First Search (BFS) and Depth-First Search (DFS) traversals starting from a given source vertex.

Input Format:
- The first line contains an integer N — the number of vertices.
- The next N lines each contain a vertex label followed by its adjacent vertices.
- The last line contains the starting vertex.

Output Format:
- Print the BFS traversal order.
- Print the DFS traversal order.

Example Input:
6
A B C
B D E
C F
D
E F
F
A

Example Output:
BFS: A B C D E F
DFS: A B D E F C

Problem 2: Shortest Path Using BFS

Given an unweighted, undirected graph, find the shortest path between two nodes using BFS traversal.

Input Format:
- The first line contains two integers N (number of nodes) and E (number of edges).
- The next E lines each contain two integers u and v indicating an edge between u and v.
- The last line contains two integers S and D — the source and destination nodes.

Output Format:
- Print the shortest path from S to D as a sequence of nodes.
- If no path exists, print 'No path exists'.

Example Input:
6 7
1 2
2 3
1 4
4 5
2 5
5 6
3 6
1 6

Example Output:
Shortest path (BFS): 1 → 2 → 5 → 6

Problem 3: Counting Connected Components (Using DFS)

Given an undirected graph, find the total number of connected components using DFS traversal.

Input Format:

- The first line contains two integers N (number of nodes) and E (number of edges).
- The next E lines each contain two integers u and v representing an undirected edge.

Output Format:

- Print the number of connected components.
- Print the list of nodes in each component.

Example Input:

6 3
1 2
3 4
5 6

Example Output:

Number of components: 3
Component 1: 1 2
Component 2: 3 4
Component 3: 5 6

Problem 4: Maze Traversal Using BFS and DFS

You are given a 2D grid representing a maze where each cell can be either free (0) or blocked (1). A robot can move up, down, left, or right from a cell. Use BFS to find the shortest path length from the top-left corner (0, 0) to a given target cell (r, c). Then, use DFS to check whether a path exists to another given cell (r2, c2).

Input Format:
- The first line contains two integers n and m (rows and columns).
- The next n lines each contain m integers (0 or 1).
- The last line contains two pairs of integers: (r, c) for BFS target and (r2, c2) for DFS target.

Output Format:
- Print the shortest path length from (0, 0) to (r, c) using BFS.
- Print whether a path exists from (0, 0) to (r2, c2) using DFS.

Example Input:
4 4
0 1 0 0
0 0 1 0
1 0 0 0
0 0 0 1
3 2 3 3

Example Output:
BFS shortest path length to (3,2): 6
DFS path to (3,3): Not found