30/4/25

Implementation of Depth first Search

Aim: To implement depth first search

Case Scenario:

a robotic delivery system is implemented in a smart warehouse, the warehouse modeled as a graph, where each node, represents a storage unit, and each edge represents a possible path,

The robot's movement strategy is to explore storage units by going as deep as possible before backtracking if needed. The warehouse is not fully mapped. So the robot uses a depth first search method. to Explore the paths.

Procedure

Step 1: Input the Graph.

- Represent the warehouse as a graph.
- Define the startnode and the goal node.

Step 2: Initialise DFS.

- use a set to track visited nodes.
- use a list to store the current traversal path.

```
# Explore neighbours
for neighbour in graph[start]:
    if neighbour not in visited:
        result = dfs(graph, neighbour, goal,
                     visited, path[:])
    if result:
        return result
return None

# Example Usage
start_node = 'A'
goal_node = 'F'
path_found = dfs(warehouse_graph,
                 start_node, goal_node)
print(f"DFS Path from {start_node} to
        {goal_node}: {path_found}")
```

Output

DFS Path from A to F : ['A', 'B', 'E', 'F']

Name of The Experiment	SYSTEM Verification	Manual Verification	Marks.
Depth First Search	1	1	2

Step 3 : Recursive DFS Function.

- Mark the current node as visited
- Add the current node to path
- Check if the current node is the goal.
  - if yes, return the path
  - if no, proceed with the next steps.

- Explore all neighbouring nodes.
  - If a neighbour is not visited, recursively call DFS on it.
  - if a path is found return it

- If No path is found return None

Step 4 : Call the DFS function.

- Call DFS with the given start and goal node
- Print the path, if found any

Result: Thus the implementation of depth first search was done successfully.