DFS

This code implements the Depth-First Search (DFS) algorithm using a stack and a graph represented as an adjacency list.

1. **Graph Representation**: The graph is represented as a dictionary where each key is a node, and the corresponding value is a list of its neighbors. For example, node '5' has neighbors '3' and '7'.

2. Visited List: The `visited` list keeps track of the nodes that have already been visited during the DFS traversal to avoid revisiting them.

3. Stack: The `stack` is used to manage the nodes to be explored. Nodes are added to the stack as they are discovered and removed as they are processed.

4. DFS Function: The `dfs` function performs the DFS traversal:

* It starts by adding the `start` node to the stack.
* It enters a loop that continues until the stack is empty.
* In each iteration, it pops a node from the stack.
* If the node has not been visited, it prints the node, adds it to the `visited` list, and checks if it is the `goal` node.
* If the node is the goal, the loop breaks.
* Otherwise, it adds all unvisited neighbors of the node to the stack.

5. **Execution**: The code prints "DFS" and then calls the `dfs` function with the graph, starting node '5', and goal node '3'.

The DFS traversal will visit nodes in the order: 5, 7, 8, 3, 4, 2, and stop when it reaches the goal node '3'.