**Depth-First Search (DFS)**

In this program we have implement the Depth-First Search (DFS) algorithm using a stack. It is done as following steps:

* + FirstImports the deque as d from the collection module for stack operations.
  + Define the DFS function that searches for goal State starting from start.
  + Initializes a stack (LIFO) with the starting node.
  + Loops until the stack is empty (i.e., all nodes are visited).
  + Removes and retrieves the last added node (DFS uses LIFO).
  + Ensures the node is not visited before processing it.
  + Marks the current node as visited.
  + If the goal state is found, the function stops searching.
  + Reverses the child nodes list before pushing them to maintain correct DFS order.
  + Adds unvisited neighbors to the stack for exploration.

Our output will be shown as one depth wise because that’s the method used in stack which goes in a depth and then back track to explore the other nodes.