**TMW1: DEPTH FIRST ITERATIVE DEEPENING SEARCH (DFID)**

from collections import defaultdict

graph = defaultdict(list)

def addEdge(u, v):

graph[u].append(v)

def dfs(start, goal, depth):

print(start, end=" ")

if start == goal:

return True

if depth <= 0:

return False

for i in graph[start]:

if dfs(i, goal, depth - 1):

return True

return False

def dfid(start, goal, maxDepth):

print("Start node: ", start, "Goal node: ", goal)

for i in range(maxDepth):

print("\nDFID at level : ", i + 1)

print("Path Taken : ", end=' ')

isPathFound = dfs(start, goal, i)

if isPathFound:

print("\nGoal node found!")

return

else:

print("\nGoal node not found!")

goal = defaultdict(list)

addEdge('A', 'B')

addEdge('A', 'C')

addEdge('A', 'D')

addEdge('B', 'E')

addEdge('B', 'F')

addEdge('E', 'I')

addEdge('E', 'J')

addEdge('D', 'G')

addEdge('D', 'H')

addEdge('G', 'K')

addEdge('G', 'L')

dfid('A', 'L', 4)

**OUTPUT:**

Start node: A Goal node: L

DFID at level : 1

Path Taken : A

DFID at level : 2

Path Taken : A B C D

DFID at level : 3

Path Taken : A B E F C D G H

DFID at level : 4

Path Taken : A B E I J F C D G K L

Goal node found!