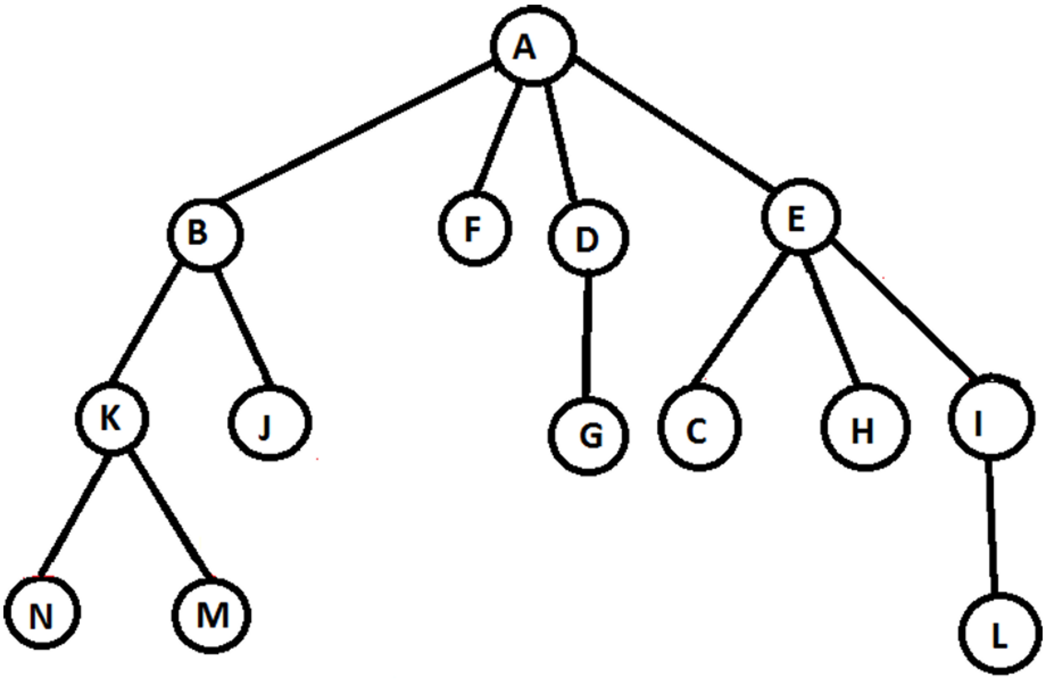
# LAB 6:

# Depth-First Search



You are required implement the following graph in python and then apply the following search

a) Depth First Search using Stack.

```
def DFS(graph, start):
    stack = []          # use a stack instead of queue
    visited = set()
    ordered_traversal = []

    stack.append(start)    # push start node into stack
    visited.add(start)

    while stack:
        node = stack.pop()   # LIFO: take the last inserted node
        print(f'Exploring {node}...')
        ordered_traversal.append(node)

        # Add neighbors (reverse order to mimic recursion-like behavior)
        for neighbour in reversed(graph[node]):
            if neighbour not in visited:
                visited.add(neighbour)
                stack.append(neighbour)

    print(f'Visited in DFS : {ordered_traversal}')

# Example graph
graph = {
    'A': ['B', 'F', 'D,'E'],
    'B': ['K', 'J'],
    'K': ['N', 'M'],
    'F': [],
    'D': ['G'],
    'E': ['C,'H','I'],
    'I': ['L']
}
```

```
start_node = 'A'

print(f"DFS traversal starting from node '{start_node}':")

DFS(graph, start_node)
```