

CODE :

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

def dfs(graph, start, max_depth):
    for depth_limit in range(max_depth + 1 ):
        print(f"Iterative Deepening Search with depth limit:
{depth_limit}")
        if dfs_recursive(graph, start, depth_limit, set()):
            return

def dfs_recursive(graph, node, depth_limit, visited):
    if node not in visited:
        print(node)
        visited.add(node)
        if depth_limit == 0:
            return False
        if node not in graph:
            return False
        for neighbor in sorted(graph[node], reverse=True):
            if dfs_recursive(graph, neighbor, depth_limit - 1, visited):
                return True
    return False

print( "Iterative Deepening Search:")
dfs(graph, 'A', 2)
```

OUTPUT :

```
● student_user@gub:~$ /bin/python3 "/home/student_user/Documents/Dulal-213902116/Lab Manual/Al lab/lab 5 13-03-24/213902116 IDS COD
E SOLVE.py"
Iterative Deepening Search:
Iterative Deepening Search with depth limit: 0
A
Iterative Deepening Search with depth limit: 1
A
D
C
B
Iterative Deepening Search with depth limit: 2
A
D
J
H
C
G
B
F
E
Iterative Deepening Search with depth limit: 3
A
D
J
P
H
O
N
C
G
M
B
F
E
L
K
○ student_user@gub:~$
```