Rollno: 5117060

DEPTH FIRST SEARCH

CODE:

```
def dfs(graph, current, goal, path, time):
    time = time + 1
    path.append(current)
    if (current == goal):
        return ("found", path, time)
    if(current in graph.keys()):
        for i in graph[current]:
            temp = dfs(graph, i, goal, path, time)
            if (temp[0] == "found"):
                return (temp)
    return ("Not found", [-1], time)
if __name__ == '__main__':
        graph = {0: 9, 9: [7, 3],
                 7: [5, 1], 3: [2, 4]}
        root = 0
        print("Graph is:", graph)
        print("To find 1")

        res, path, time = dfs(graph, graph[root], 1, [], 0)
        print(res)
        print("Path is",path)
        print("N = 7 , b = 2 , d = 2")
        print("Time Complexity = O(", len(path), ")")
        print("Space Complexity = O(", time, ")")
```

OUTPUT:

```
OUTPUT: Graph is: {0: 9, 9: [7, 3], 7: [5, 1], 3: [2, 4]}
To find 1
found
Path is [9, 7, 5, 1]
N = 7 , b = 2 , d = 2
Time Complexity = O( 4 )
Space Complexity = O( 3 )
```