```
Print (“******************DFS****************”)
Class Network:
    def __init__(self):
        Self.adj_list = {}

    def add_connection(self, src, dest):
        If src not in self.adj_list:
            Self.adj_list[src] = []
        Self.adj_list[src].append(dest)

    def depth_first_search(self, start, target):
        Explored = set()
        To_visit = [start]

        While to_visit:
            Current = to_visit.pop()
            If current == target:
                Print(f”Target node {target} found!”)
                Return True
            If current not in explored:
                Print(current)
                Explored.add(current)
                For neighbor in self.adj_list.get(current, []):
                    If neighbor not in explored:
                        To_visit.append(neighbor)

        Print(f”Target node {target} not found.”)
        Return False
```

```
Net = Network()
Net.add_connection('A', 'B')
Net.add_connection('A', 'C')
Net.add_connection('B', 'D')
Net.add_connection('B', 'E')
Net.add_connection('C', 'F')
Net.add_connection('D', 'G')
Net.add_connection('E', 'G')

Print("DFS starting from node A:")
Net.depth_first_search('A', 'G')
```