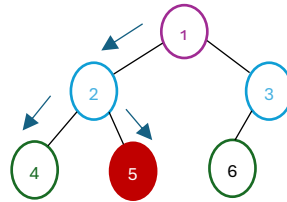


## Assignment 2

# DFS

## Using Python Class



```
class Graph:
    def __init__(self):
        self.graph = {}
    def add_node(self, node):
        if node not in self.graph:
            self.graph[node] = []
    def add_edge(self, node1, node2):
        if node1 in self.graph and node2 in self.graph:
            self.graph[node1].append(node2)
            self.graph[node2].append(node1)
def DFS(visited, graph, node, target):
    if node not in visited:
        print(node, end=' ')
        visited.add(node)
        if node == target:
            return True
        for neighbor in graph[node]:
            if neighbor not in visited:
                if DFS(visited, graph, neighbor, target):
                    return True
    return False
graph=Graph()
graph.add_node('1')
graph.add_node('2')
graph.add_node('3')
graph.add_node('4')
graph.add_node('5')
graph.add_node('6')

graph.add_edge('1','2')
graph.add_edge('1','3')
graph.add_edge('2','4')
graph.add_edge('2','5')
graph.add_edge('3','6')
visited = set()
DFS(visited,graph.graph,'1','5')
```

1 2 4 5

True

Output