

```

In [1]: # BFS
graph = {'A':['B', 'E', 'C'],
         'B':['A', 'D', 'E'],
         'D':['B', 'E'],
         'E':['A', 'D', 'B'],
         'C':['A', 'F', 'G'],
         'F':['C'],
         'G':['C']}

visited = []
queue = []

def bfs(visited, graph, start_node, goal_node):
    visited.append(start_node)
    queue.append(start_node)
    while queue:
        m = queue.pop(0)
        print(m)
        if m == goal_node:
            print("Node is Found !!! ")
            break
        else:
            for n in graph[m]:
                if n not in visited:
                    visited.append(n)
                    queue.append(n)

print("The BFS Traversal is : ")
bfs(visited, graph, 'A', 'D')

#DFS
graph = {'A':['B', 'E', 'C'],
         'B':['A', 'D', 'E'],
         'D':['B', 'E'],
         'E':['A', 'D', 'B'],
         'C':['A', 'F', 'G'],
         'F':['C'],
         'G':['C']}

visited = []
stack = []
def dfs(graph, start, goal):
    print("DFS traversal is: ")
    stack.append(start)
    visited.append(start)
    while stack:
        node = stack[-1]
        stack.pop()
        print("Node: ", node)
        if node == goal:
            print("Goal node found!")
            return
        for n in graph[node]:
            if n not in visited:
                visited.append(n)
                stack.append(n)

```

```
dfs(graph, 'A', "D")
```

The BFS Traversal is :

A

B

E

C

D

Node is Found !!!

DFS traversal is:

Node: A

Node: C

Node: G

Node: F

Node: E

Node: D

Goal node found!

In []: