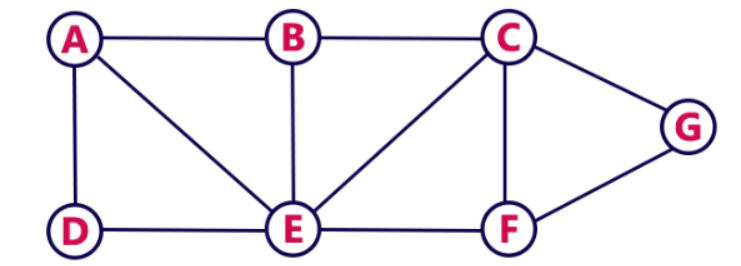
**22AIE203- DATASTRUCTURES & ALGORITHMS 2**

**LABSHEET 1**

Name: Anuvind M P

Roll no: AM.EN.U4AIE22010

--------------------------------------------------------------------------------------------------------------------------



1. **Breadth First Search :**

visited = []

N = ["A","B","C","D","E","F","G"]

Q = []

Graph = [

    [0,1,0,1,1,0,0],

    [1,0,1,0,1,0,0],

    [0,1,0,0,1,1,1],

    [1,0,0,0,1,0,0],

    [1,1,1,1,0,1,0],

    [0,0,1,0,1,0,1],

    [0,0,1,0,0,1,0]

]

def Dequeue(Q):

    Q=Q[1:]

    return Q

def BFS\_Search(Q):

    for v in N:

        if v not in visited:

            BFS(Q,v)

def BFS(Q,v):

    visited.append(v)

    Q.append(v)

    while len(Q) != 0:

        u = Q[0]

        Q = Dequeue(Q)

        pos = 0

        adj = Graph[N.index(u)]

        for i in adj:

            if i:

                w = N[pos]

                if w not in visited:

                    visited.append(w)

                    if w not in Q:

                        Q.append(w)

            pos+=1

BFS\_Search(Q)

print(visited)

**Output:**



**2. Depth First Search:**

visited = []

N = ["A","B","C","D","E","F","G"]

Graph = [

    [0,1,0,1,1,0,0],

    [1,0,1,0,1,0,0],

    [0,1,0,0,1,1,1],

    [1,0,0,0,1,0,0],

    [1,1,1,1,0,1,0],

    [0,0,1,0,1,0,1],

    [0,0,1,0,0,1,0]

]

def DFS\_Search():

    for v in N:

        if v not in visited:

            DFS(v)

def DFS(v):

    visited.append(v)

    pos = 0

    adj = Graph[N.index(v)]

    for i in adj:

        if i:

            w = N[pos]

            if w not in visited:

                DFS(w)

        pos+=1

DFS\_Search()

print(visited)

**Output:**

