

BFS

```
In [1]: graph={
    'A':['B', 'C'],
    'B':['D', 'E'],
    'C':['F'],
    'D':[],
    'E':[],
    'F':[]
}
visited=[]
queue=[]
def bfs(visited,graph,node):
    visited.append(node)
    queue.append(node)
    while queue:
        s=queue.pop(0)
        print(s,end=" ")

        for neighbour in graph[s]:
            if neighbour not in visited:
                visited.append(neighbour)
                queue.append(neighbour)
print('Following is the Path using Breadth-First-Search')
bfs(visited,graph,'A')
```

Following is the Path using Breadth-First-Search
A B C D E F

DFS

```
In [2]: graph={
    'A':['B', 'C'],
    'B':['D', 'E'],
    'C':['F'],
    'D':[],
    'E':[],
    'F':[]
}
visited=set()
def dfs(visited,graph,node):
    if node not in visited:
        print(node,end=" ")
        visited.add(node)
        for neighbour in graph[node]:
            dfs(visited,graph,neighbour)
print('Following is the Path using Depth-First-Search')
dfs(visited,graph,'A')
```

Following is the Path using Depth-First-Search
A B D E C F

In []: