

1. Write a python program to show how BFS & DFS work on the search tree for given state space graph.



Program :

```
from collections import deque
```

```
# Define the graph as an adjacency list
```

```
graph = {  
    'A': ['B', 'C'],  
    'B': ['D'],  
    'C': ['E', 'F'],  
    'D': [],  
    'E': [],  
    'F': []  
}
```

```
def bfs(graph, start):
```

```
    visited = set()
```

```
    queue = deque([start])
```

```
    while queue:
```

```
        node = queue.popleft()
```

```
        if node not in visited:
```

```
print(node, end=' ')
visited.add(node)
queue.extend(graph[node])
```

```
def dfs(graph, start):
```

```
    visited = set()
```

```
    def dfs_recursive(node):
```

```
        nonlocal visited
```

```
        if node not in visited:
```

```
            print(node, end=' ')
```

```
            visited.add(node)
```

```
            for neighbor in graph[node]:
```

```
                dfs_recursive(neighbor)
```

```
    dfs_recursive(start)
```

```
# Perform BFS and DFS on the given graph
```

```
print("BFS Traversal:")
```

```
bfs(graph, 'A')
```

```
print("\nDFS Traversal:")
```

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

Output :

BFS Traversal:

A B C D E F

DFS Traversal:

A B D C E F