BFS and DFS Algorithms

Breadth-First Search (BFS) Algorithm

```
from collections import deque


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(neighbor for neighbor in graph[node] if neighbor not in visited)


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


print("BFS Traversal:")
```

BFS and DFS Algorithms

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

Depth-First Search (DFS) Algorithm

```
def dfs(graph, node, visited=set()):
    if node not in visited:
        print(node, end=' ')
        visited.add(node)
        for neighbor in graph[node]:
            dfs(graph, neighbor, visited)


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


print("DFS Traversal:")
dfs(graph, 'A')
```