from collections import deque

def bfs(graph, start):

queue, visited = deque([start]), set()

print("BFS:", end=" ")

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)

print()

def dfs(graph, start):

stack, visited = [start], set()

print("DFS:", end=" ")

while stack:

node = stack.pop()

if node not in visited:

print(node, end=" ")

visited.add(node)

stack.extend(reversed([neighbor for neighbor in graph[node] if neighbor not in visited]))

print()

# Example graph

graph = {

'A': ['B', 'C'],

'B': ['A', 'D', 'E'],

'C': ['A', 'F'],

'D': ['B'],

'E': ['B', 'F'],

'F': ['C', 'E']

}

bfs(graph, 'A')

dfs(graph, 'A')