# DFS

def dfs(graph, start, visited):

visited.add(start)

for neighbor in graph[start]:

if neighbor not in visited:

dfs(graph, neighbor, visited)

# BFS

def bfs(graph, start):

visited = set([start])

queue = [start]

while queue:

node = queue.pop(0)

for neighbor in graph[node]:

if neighbor not in visited:

visited.add(neighbor)

queue.append(neighbor)

# Example

graph = {

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

'B': ['D'],

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

'D': [],

'E': [],

'F': []

}

# DFS

visited = set()

dfs(graph, 'A', visited)

print(visited)

# BFS

visited = set()

queue = ['A']

while queue:

node = queue.pop(0)

print(node)

for neighbor in graph[node]:

if neighbor not in visited:

visited.add(neighbor)

queue.append(neighbor)