**Assignment No : 2**

**Title: Implement depth first search and breadth first search algorithm. Use an undirected graph and develop a recursive algorithm for searching all the vertices of the graph or tree data structure.**

from collections import deque

def bfs(graph, start):

visited = set()

queue = deque([start])

while queue:

node = queue.popleft()

if node not in visited:

visited.add(node)

print(node)

for neighbor in graph[node]:

if neighbor not in visited:

queue.append(neighbor)

def dfs(graph, start, visited=None):

if visited is None:

visited = set()

visited.add(start)

print(start)

for neighbor in graph[start]:

if neighbor not in visited:

dfs(graph, neighbor, visited)

# Example usage:

graph = {

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

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

'C': ['A', 'E'],

'D': ['B'],

'E': ['C']

}

print("BFS:")

bfs(graph, 'A')

print("DFS:")

dfs(graph, 'A') **OUTPUT:**

BFS:

A

B

C

D

E

DFS:

A

B

D

C

E

**Time complexity:** DFS: O(b^m)

BFS: O(V+E)