**Implementation and analysis of DFS and BFS for an application**

**CODE:**

graph = {

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

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

'C' : ['F'],

'D' : [],

'E' : ['F'],

'F' : []

}

visited\_bfs = []

queue = []

def bfs(visited\_bfs, graph, node):

visited\_bfs.append(node)

queue.append(node)

while queue:

s = queue.pop(0)

print (s, end = " ")

for neighbour in graph[s]:

if neighbour not in visited\_bfs:

visited\_bfs.append(neighbour)

queue.append(neighbour)

visited = set()

def dfs(visited, graph, node):

if node not in visited:

print (node, end=" ")

visited.add(node)

for neighbour in graph[node]:

dfs(visited, graph, neighbour)

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

bfs(visited\_bfs, graph, 'A')

print('\n')

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

dfs(visited, graph, 'A')