**LP- II (AI)-ASSIGNMENT – 01**

**Om Jadhav (21CO050)**

**(BFS AND DFS)**

**PROGRAM**

from collections import defaultdict

class Graph:

def \_\_init\_\_(self):

self.graph = defaultdict(list)

def add\_edge(self, u, v):

self.graph[u].append(v)

def dfs\_util(self, v, visited):

visited[v] = True

print(v, end=' ')

for i in self.graph[v]:

if not visited[i]:

self.dfs\_util(i, visited)

def dfs(self, start):

visited = [False] \* (max(self.graph) + 1)

self.dfs\_util(start, visited)

print()

def bfs(self, start):

visited = [False] \* (max(self.graph) + 1)

queue = []

queue.append(start)

visited[start] = True

while queue:

v = queue.pop(0)

print(v, end=' ')

for i in self.graph[v]:

if not visited[i]:

queue.append(i)

visited[i] = True

print()

def main():

g = Graph()

while True:

print("\n1. Add Edge")

print("2. Depth First Search (DFS)")

print("3. Breadth First Search (BFS)")

print("4. Quit")

choice = input("Enter your choice: ")

if choice == '1':

u = int(input("Enter source vertex: "))

v = int(input("Enter destination vertex: "))

g.add\_edge(u, v)

elif choice == '2':

start = int(input("Enter starting vertex for DFS: "))

print("DFS traversal:")

g.dfs(start)

elif choice == '3':

start = int(input("Enter starting vertex for BFS: "))

print("BFS traversal:")

g.bfs(start)

elif choice == '4':

print("Exiting...")

break

else:

print("Invalid choice! Please choose a valid option.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**OUTPUT**

1. Add Edge

2. Depth First Search (DFS)

3. Breadth First Search (BFS)

4. Quit

Enter your choice: 1

Enter source vertex: 0

Enter destination vertex: 1

1. Add Edge

2. Depth First Search (DFS)

3. Breadth First Search (BFS)

4. Quit

Enter your choice: 1

Enter source vertex: 0

Enter destination vertex: 2

1. Add Edge

2. Depth First Search (DFS)

3. Breadth First Search (BFS)

4. Quit

Enter your choice: 1

Enter source vertex: 1

Enter destination vertex: 2

1. Add Edge

2. Depth First Search (DFS)

3. Breadth First Search (BFS)

4. Quit

Enter your choice: 1

Enter source vertex: 2

Enter destination vertex: 0

1. Add Edge

2. Depth First Search (DFS)

3. Breadth First Search (BFS)

4. Quit

Enter your choice: 1

Enter source vertex: 2

Enter destination vertex: 3

1. Add Edge

2. Depth First Search (DFS)

3. Breadth First Search (BFS)

4. Quit

Enter your choice: 1

Enter source vertex: 3

Enter destination vertex: 3

1. Add Edge

2. Depth First Search (DFS)

3. Breadth First Search (BFS)

4. Quit

Enter your choice: 2

Enter starting vertex for DFS: 1

DFS traversal:

1 2 0 3

1. Add Edge

2. Depth First Search (DFS)

3. Breadth First Search (BFS)

4. Quit

Enter your choice: 3

Enter starting vertex for BFS: 1

BFS traversal:

1 2 0 3

1. Add Edge

2. Depth First Search (DFS)

3. Breadth First Search (BFS)

4. Quit

Enter your choice: 2

Enter starting vertex for DFS: 2

DFS traversal:

2 0 1 3

1. Add Edge

2. Depth First Search (DFS)

3. Breadth First Search (BFS)

4. Quit

Enter your choice: 3

Enter starting vertex for BFS: 2

BFS traversal:

2 0 3 1

1. Add Edge

2. Depth First Search (DFS)

3. Breadth First Search (BFS)

4. Quit

Enter your choice: 4

Exiting...