**AI Assignment 1**

**Neeti Kurulkar**

**DFS BFS algorithm**

**Code:**

**class** **Graph:**

**def** \_\_init\_\_**(**self**,** v**):**

self**.***v* **=** v

self**.***adj\_mat* **=** **[[**0 **for** \_ **in** **range(**v**)]** **for** \_ **in** **range(**v**)]**

**def** add\_edge**(**self**,** a**,** b**):**

self**.***adj\_mat***[**a**][**b**]** **=** 1

self**.***adj\_mat***[**b**][**a**]** **=** 1

**def** dfs**(**self**,** node**,** visited**):**

"""Recursive Depth First Search (DFS)."""

**if** node **not** **in** visited**:**

**print(**node**,** end**=**' '**)**

visited**.***add***(**node**)**

**for** neighbor **in** **range(**self**.***v***):**

**if** self**.***adj\_mat***[**node**][**neighbor**]** **!=** 0 **and** neighbor **not** **in** visited**:**

self**.***dfs***(**neighbor**,** visited**)**

**def** bfs**(**self**,** start**):**

"""Breadth First Search (BFS)"""

visited **=** **set()**

queue **=** **[**start**]**

**while** queue**:**

node **=** queue**.***pop***(**0**)**

**if** node **not** **in** visited**:**

**print(**node**,** end**=**' '**)**

visited**.***add***(**node**)**

**for** neighbor **in** **range(**self**.***v***):**

**if** self**.***adj\_mat***[**node**][**neighbor**]** **!=** 0 **and** neighbor **not** **in** visited**:**

queue**.***append***(**neighbor**)**

n **=** **int(input(**"Enter number of vertices: "**))**

g **=** Graph**(**n**)**

e **=** **int(input(**"Enter number of edges: "**))**

**print(**"Enter edges (u v):"**)**

**for** \_ **in** **range(**e**):**

u**,** v **=** **map(int,** **input().***split***())**

g**.***add\_edge***(**u**,** v**)**

start\_node **=** **int(input(**"Enter the starting node for traversal: "**))**

**print(**"DFS:"**)**

g**.***dfs***(**start\_node**,** **set())**

**print(**"\nBFS:"**)**

g**.***bfs***(**start\_node**)**

**Output:**

