Data Structures Lab (CSC204) **Session: Monsoon 2023-24**
III BTech (CSE) **Date: 30.10.2023**

Assignment-9 (Graph-II)

---

1. A directed graph is represented as $G = (V, E)$, where V represents the set of vertices and E shows the set of directed edges. A graph could be implemented either using adjacency matrix or adjacency list. Perform the following tasks for the adjacency list representation of a directed graph.

a. Write a function ***find_degree***( ) to print the *in-degree* and *out-degree* of all the vertices in the given directed graph.

Sample Input:
Enter the adjacency list:
A→B→C→D→F
B→E
D→B→E
E→C→G
F→D

Sample Output:
A: 0, 4
B: 2, 1
C: 2, 0
D: 2, 2
E: 2, 2
F: 1, 1
G: 1, 0

b. Write a function ***is_path***( ) that takes the adjacency list of a directed graph, as well as two vertices, and outputs whether there is a path from the first vertex to the second vertex. If a path exists, the function should also print the path itself.

Sample Input:
Enter the source vertex: A
Enter the destination vertex: G

Sample Output:
Path exists from A to G and the path is A→B→E→G.

c. Write a function ***BFS_ traversal*** ( ) to traverse the above graph using BFS from a given source vertex. Use alphabetical sorted order to break the tie, if any.

Sample Input:
Enter the source vertex: A

Sample Output:
The BFS sequence starting from vertex A is: A → B → C → D → F → E → G

d. Write a function ***Topo_Sort***( ) to perform DFS-based topological sort of the vertices of the directed graph generated in Q.1a from a given source vertex. Use alphabetical sorted order to break the tie, if any.

Sample Input:
Enter the source vertex: A

Sample Output: A → F → D → B →E → G → C