# Aror University of Art, Architecture, Design & Heritage Sukkur

Department of AI-Multimedia and Gaming

BS-AI (Sec# A and B), Fall 2024 Batch

---

Lab 11: Graph Representation and Operations Using Adjacency Matrices

Subject: Data Structure (CSC221) Date: 03 December, 2024 Instructor: Abdul Ghafoor

---

## Objectives:

1. Understand graph representation using adjacency matrices.
2. Learn how to perform graph operations programmatically.
3. Solve real-world graph problems like finding the shortest path.

**Task 1: Implement a Graph Using an Adjacency Matrix**

Objective: Create and represent a graph using an adjacency matrix.

- Input: Number of vertices and edges.
- Input the edges as `(u, v)` pairs, where `u` and `v` are the connected vertices.
- Output: Display the adjacency matrix of the graph.

*Example Input:*
Vertices: 4
Edges: [(1, 2), (2, 3), (3, 4), (4, 1)]

*Example Output:*

```
Adjacency Matrix:

0 1 0 0

0 0 1 0

0 0 0 1

1 0 0 0
```

**Task 2: Display the Graph as an Adjacency Matrix**

Objective: Extend Task 1 by creating a function to display the adjacency matrix in a readable format.

- Print the adjacency matrix row by row.
- Ensure the output clearly identifies rows and columns corresponding to vertices.

**Task 3: Find the Shortest Path Between Two Vertices**

Objective: Using the adjacency matrix, implement a function to compute the shortest path between any two vertices.

- Input: Starting vertex and destination vertex.
- Output: The shortest path length and the sequence of vertices.

Start: 1, Destination: 4

*Example Output:*
Shortest Path: 1 → 2 → 4
Length: 2

**Task 4: Find All Paths Between Two Vertices**

Objective: Implement a function to find all possible paths between two vertices in the graph.

- Input: Starting and destination vertices.
- Output: List all paths and their respective lengths.

Start: 1, Destination: 4

*Example Output:*
Paths:
1 → 2 → 4 (Length: 2)
1 → 3 → 4 (Length: 2)

**Task 5: Detect Connected Components in an Undirected Graph**

Objective: Identify all connected components of an undirected graph using adjacency matrices.

- Input: Adjacency matrix of an undirected graph.
- Output: List all connected components (group of vertices).

*Example Output:*

Connected Components:

Component 1: {1, 2, 3}

Component 2: {4, 5}

# Repeat all above tasks Weighted Graph Using an Adjacency Matrix