



**DEPARTMENT OF COMPUTER & SOFTWARE  
ENGINEERING**  
**COLLEGE OF E&ME, NUST, RAWALPINDI**



# EC-350 Artificial Intelligence and Decision Support System

## LAB MANUAL – 03

**Course Instructor:** Assoc Prof Dr Arsalan Shaukat

**Lab Engineer:** Kashaf Raheem

**Student Name:** \_\_\_\_\_

**Degree/ Syndicate:** \_\_\_\_\_

## **LAB # 3: GRAPH THEORY & PATH SEARCHES IN PYTHON**

### **Lab Objective:**

- To familiarize students with the fundamentals of graph theory and path searches in Python.

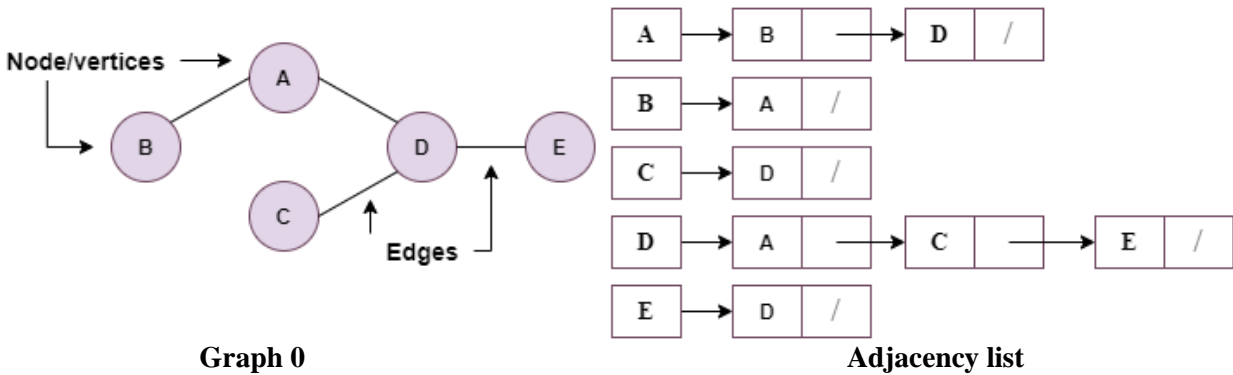
### **Hardware/Software required:**

Hardware: Desktop/ Notebook Computer

Software Tool: Python 3.10.0

### **Lab Description:**

Graph theory is the fundamental concept in mathematics, and it lays the foundation of many algorithms in the field of networking, artificial intelligence, and image processing. Graph theory is the study of graphs, which are mathematical structures used to model pairwise relations between objects. A graph in this context is made up of vertices, nodes, or points which are connected by edges, arcs, or lines. A graph may be undirected, meaning that there is no distinction between the two vertices associated with each edge, or its edges may be directed from one vertex to another. A Graph can be represented as a collection of linked lists known as adjacency list. It can also be stored in form of 2-D matrix known as adjacency matrix.



	A	B	C	D	E
A	0	1	0	1	0
B	1	0	0	0	0
C	0	0	0	1	0
D	1	0	1	0	1
E	0	0	0	1	0

**Adjacency Matrix**

In Python, graphs can be easily implemented through key-value paired data structure i.e. dictionaries.

**Example:**

Graph0= {

B: [A],

A: [B, D],

C: [D],

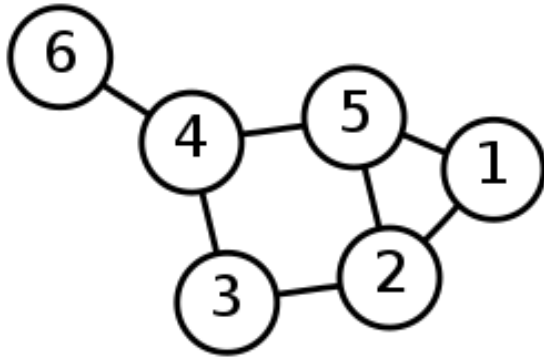
D: [A, C, E],

E: [D]

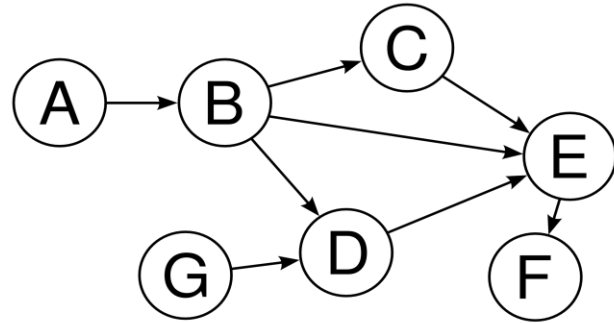
}

**Lab Task:****Q1:**

- a. Implement the undirected Graph 1 in Python. Show the connectivity as well as the degree of each node within these graphs.
- b. Implement the directed Graph 2 in Python. Show the connectivity, indegree and outdegree of each node within these graphs.
- c. Write a method to find any path between node 6 to node 1 in Graph 1.
- d. Write a method to find any path between node A to node F in Graph 2.
- e. Modify Task c to show all possible paths between node 6 to node 1 in Graph 1.
- f. Modify Task f to show all possible paths between node A to node F in Graph 2.
- g. Represent Graph 1 and Graph 2 by adjacency list.



Graph 1



Graph 2

**Q2:** Suppose you have been given with a following 3x3 3-bit grayscale image. Your job is to decompose it into an undirected graph where each pixel within an image represent a node and adjacent nodes are connected to each other via 4-connectivity pattern. Show all possible paths between pixel 150 and pixel 165.

150	2	5
80	145	45
74	102	165

**Bonus Question:**

**Q:** Decompose the above image into an undirected graph where each pixel represents a node and the edge cost between adjacent nodes is computed by taking the absolute difference. Now segment the object between node 150 to node 165 by computing the shortest path. **Hint: Use nested dictionaries to represent graph with edge costs.**

**Home Task:**

**Q1:** You have been given with following 5x5 weighted matrix of undirected graph. Where (V) represents the vertices/nodes while elements in the matrix represents weight of edges from one node to another. Your job is to implement it to undirected graph and find the shortest path from V1 to V3.

	V1	V2	V3	V4	V5
V1	0	5	$\infty$	6	$\infty$
V2	5	0	9	$\infty$	7
V3	0	9	0	$\infty$	9
V4	6	$\infty$	$\infty$	0	5
V5	$\infty$	7	9	5	0