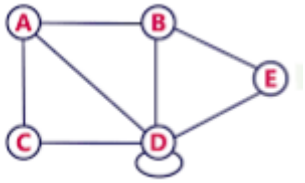
Department of Computer Science and Engineering,
Anna University, Chennai- 600025
CS23301 – Data Structures (R2023)
Practical
July – November 2026
Year/Sem/Batch : II/III/ N

Exercise: 11	GRAPHS AND SPANNING TREE	09 – Oct - 2026

Observation (5 Marks)

1. List the different types of graphs
2. What are the two ways to represent graphs?
3. Represent the following graph in adjacency matrix and adjacency list.



4. Write 5 BFS and DFS traversal for the above graph.
5. How would you built a Netflix recommendation algorithm using graph modelling? What would you be your nodes?. What would represent the edges? What additional elements can you use to improve your recommendation system?
6. The adjacency matrix of a graph is given below.

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

(a) Draw the graph defined by this adjacency matrix. Label the vertices of your graph so the vertex i corresponds to row and column j of the matrix.

(b) In a graph, we define the distance between to vertices to be the length of the shortest path between them. We define the diameter of a graph to be the largest distance between any two nodes. What is the diameter of this graph?. Explain why.

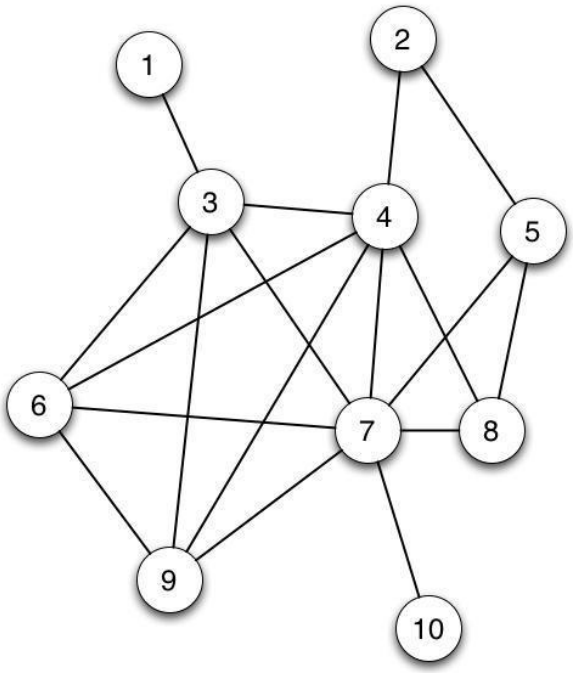
7. The following table represents the set of delivery points where you must go in one of your delivery days for Amazon. Together with the delivery points you can find information about the distance between them. Make a graphical representation of the connection networks and provide also the adjacent and incident matrix.

Packages (delivery points)
1. Barcelona
2. Badalona
3. Rubí
4. San Cugat
5. Tarrasa
6. Cerdanyola

Origin	Destination	Road
Barcelona	Badalona	3km
Barcelona	Rubí	5km
Badalona	Rubí	1km
Badalona	San Cugat	4km
Rubí	San Cugat	3km
Badalona	Cerdanyola	5km
Cerdanyola	San Cugat	8km
Tarrasa	Rubí	2km
Tarrasa	San Cugat	7km

Execution (15 Marks)

1. Implement BFS and DFS traversal for the given graph



2. Anna university has N students numbered 1 to N. Some pairs of students are known to copy from each other. Such a pair is called a rival pair, and the two students must not be placed in the same room. You are given the rival pairs. The exam cell has only two rooms: Room A and Room B. Write a program that:
   - Decides whether a valid allocation exists.
   - If yes, prints the students in each room.
   - If no, prints an odd cycle among the students, proving that no valid allocation exists.
   - The rivalry graph may be disconnected, so every component must be checked.

     Notes:
     - Model students as vertices and rival pairs as undirected edges.

- A graph is bipartite if its vertices can be 2-colored so that no edge joins two vertices of the same color.
- Theorem: A graph is bipartite if and only if it contains no odd-length cycle.
- In BFS, vertices at even levels get one color and vertices at odd levels get the other. If an edge joins two vertices at the same level parity, an odd cycle exists.

Input Format

N M

u1 v1

u2 v2

...

uM vM

- N: number of students
- M: number of rival pairs
- Each (u, v) is an undirected edge (rivalry is mutual)

Output Format

If allocation is possible:

POSSIBLE

Room A: <students in increasing order>

Room B: <students in increasing order>

If allocation is not possible:

NOT POSSIBLE

Odd cycle (length L): v1 v2 v3 ... vL

Consecutive vertices in the printed cycle must be adjacent, and vL must be adjacent to v1. L must be odd.

Constraints

- $1 \leq N \leq 10^5$

- $0 \leq M \leq 2 \times 10^5$
- $1 \leq u, v \leq N$
- The graph may contain isolated vertices and multiple components.
- Self-loops ($u = v$) may appear. A self-loop is an odd cycle of length 1.

Sample 1: Even cycle (possible)

Input:

6 6

1 2

2 3

3 4

4 5

5 6

6 1

Output:

POSSIBLE

Room A: 1 3 5

Room B: 2 4 6

Sample 2: Triangle (not possible)

Input:

5 5

1 2

2 3

3 1

3 4

4 5

Output:

NOT POSSIBLE

Odd cycle (length 3): 2 1 3

Any valid odd cycle is accepted, for example 1 2 3.

Sample 3: Disconnected graph, one bad component

Input:

7 5

1 2

2 3

4 5

5 6

6 4

Output:

NOT POSSIBLE

Odd cycle (length 3): 5 4 6

Here the first component (1–2–3) is bipartite, but the second (4–5–6) is a triangle. A program that checks only the component of vertex 1 will give the wrong answer.

Sample 4: No rivalries

Input:

4 0

Output:

POSSIBLE

Room A: 1 2 3 4

Room B: