Assignment - 2

Name : Soham Das

Roll : 002311001004

Department : Information Technology (A3)

A. Write a Python program to load and read a binary dataset from a CSV file and draw the corresponding graph considering the dataset as an adjacency matrix.

This part of the code reads a CSV file containing a binary adjacency matrix and visualizes it as an unweighted graph.

Objective: To load a binary dataset from a CSV file and draw the corresponding graph.

Steps:

- **Load Data:** The script reads the CSV file into a data table using the `pandas` library.
- **Create Matrix:** This table is converted into a standard numerical matrix.
- **Build Graph:** The `networkx` library interprets the matrix, creating a node for each row and connecting nodes where a '1' is present.
- **Visualize:** `matplotlib` draws the final network, showing all the nodes and their connections.

Input: `soham_binary.csv`

0	*1*	*1*	*0*	*1*	*0*	*1*	*0*
1	*0*	*1*	*0*	*0*	*0*	*0*	*1*
1	*0*	*0*	*0*	*0*	*0*	*1*	*1*
0	*0*	*1*	*0*	*1*	*0*	*1*	*0*

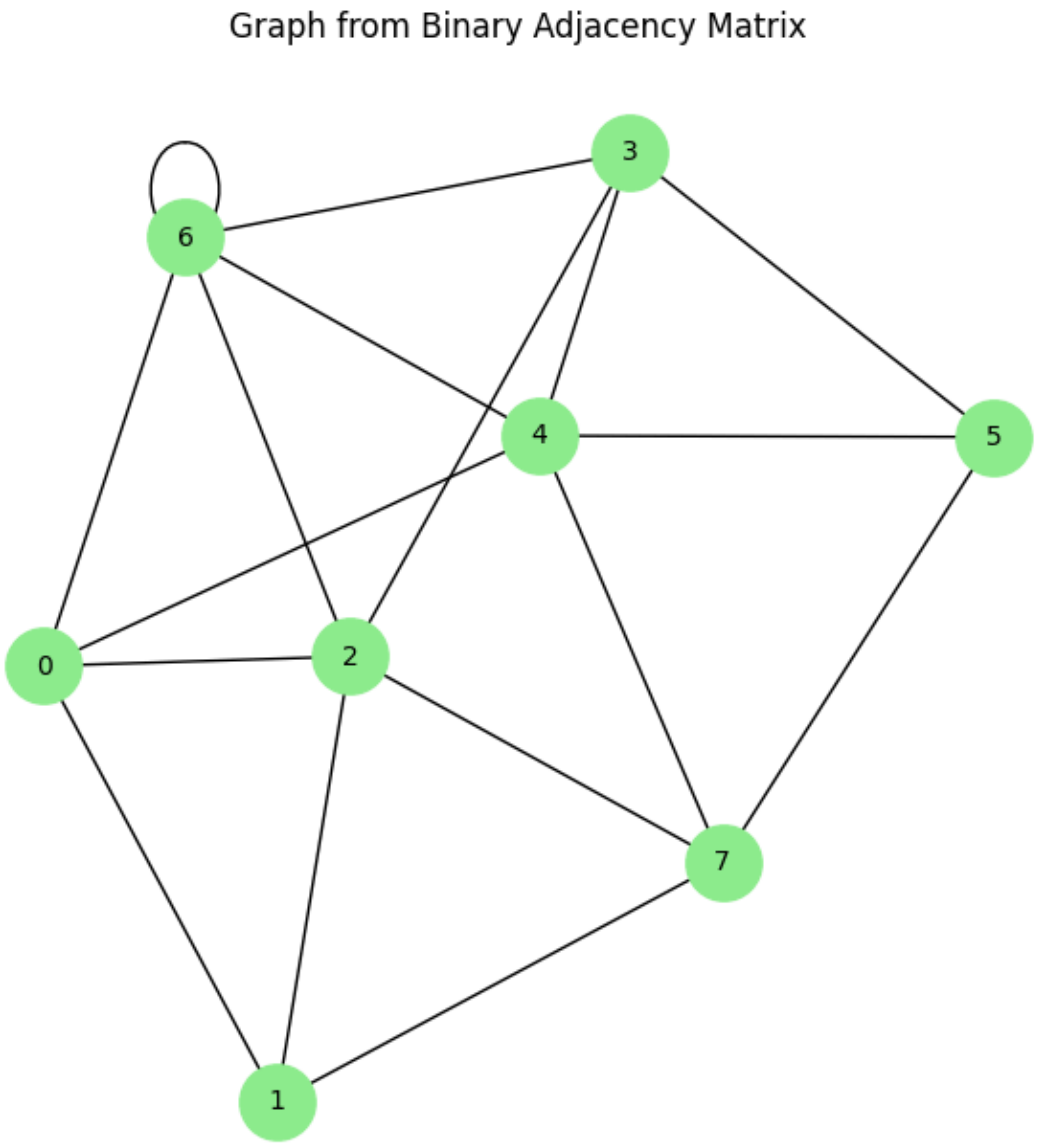
1 0 0 0 0 1 1 1

0 0 0 1 0 0 0 1

1 0 0 0 0 0 1 0

0 0 1 0 1 0 0 0

Output Graph



B. Write a Python program to load and read a numerical dataset from a CSV file and draw the corresponding graph considering the dataset as an adjacency matrix.

This part of the code reads a CSV file containing a numerical adjacency matrix and visualizing it as a weighted graph, where edge labels represent the weights.

Objective: To load a numerical dataset from a CSV file and draw the corresponding weighted graph.

Steps:

- **Load Data:** The script reads the numerical CSV file into a data table using `pandas`.
- **Create Matrix:** The table is converted into a numerical matrix.
- **Build Weighted Graph:** `networkx` builds a graph, treating non-zero values as connections and storing them as edge **'weights'**.
- **Visualize & Label:** `matplotlib` draws the network and adds the stored 'weight' values as labels on each connection line.

Input: soham_numeric.csv

0	*5*	*0*	*0*	*0*	*0*	*0*	*1*
5	*0*	*0*	*0*	*2*	*0*	*0*	*0*
0	*0*	*0*	*0*	*4*	*0*	*4*	*3*
0	*3*	*8*	*0*	*0*	*0*	*0*	*0*

0 0 3 0 1 0 0 0

5 0 0 9 0 0 7 0

0 3 5 0 0 0 0 6

1 0 0 0 5 0 0 0

Output Graph