**PSG College of Technology Department of Applied Mathematics and Computational Science M.Sc (Cyber Security) - Semester VI 20XCE8 - Social Network Analysis Problem Sheet 1**

**R. Hari haran**

**22PC14**

**1. import networkx as nx**

**import matplotlib.pyplot as plt**

**nodes = ["MIT", "HARV", "CARN", "CASE", "LINC", "BBN"]**

**adj\_matrix = [**

**[0, 0, 0, 1, 1, 0],**

**[0, 0, 1, 0, 0, 1],**

**[0, 1, 0, 1, 0, 0],**

**[0, 0, 1, 0, 1, 0],**

**[1, 0, 0, 1, 0, 0],**

**[1, 1, 0, 0, 0, 0],**

**]**

**G = nx.Graph()**

**G.add\_nodes\_from(nodes)**

**for i, node1 in enumerate(nodes):**

**for j, node2 in enumerate(nodes):**

**if i != j and adj\_matrix[i][j] == 1:**

**G.add\_edge(node1, node2)**

**plt.figure(figsize=(8, 6))**

**pos = nx.spring\_layout(G) nx.draw(**

**G,**

**pos,**

**with\_labels=True,**

**node\_color="skyblue",**

**node\_size=2000,**

**font\_size=10,**

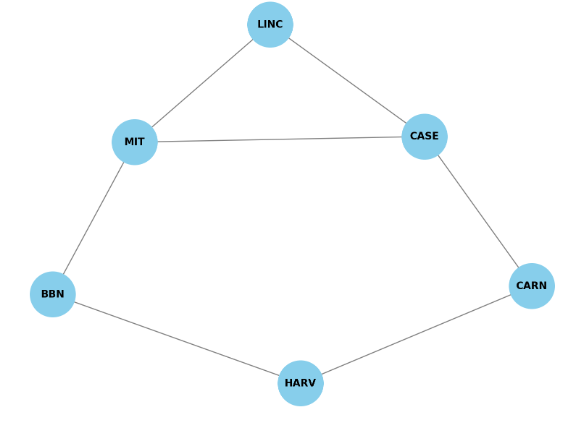
**font\_weight="bold",**

**edge\_color="gray",**

**)**

**plt.title("Graph Representation of the Adjacency Matrix", fontsize=14)**

**plt.show()**

****

2. import networkx as nx

import matplotlib.pyplot as plt

arpanet\_nodes = ["SDC", "RAND", "SRI", "UCLA", "UCSB", "IPTO", "BBN", "MIT"]

arpanet\_edges = [

("SDC", "RAND"), ("RAND", "UCLA"), ("UCLA", "UCSB"),

("UCSB", "SRI"), ("SRI", "IPTO"), ("IPTO", "BBN"),

("BBN", "MIT"), ("SDC", "UCLA"), ("UCLA", "SRI"),

("BBN", "RAND"), ("MIT", "SRI")

]

arpanet\_graph = nx.Graph()

arpanet\_graph.add\_nodes\_from(arpanet\_nodes)

arpanet\_graph.add\_edges\_from(arpanet\_edges)

vertices = list(arpanet\_graph.nodes)

edges = list(arpanet\_graph.edges)

length = arpanet\_graph.size() # Number of edges

eccentricity = nx.eccentricity(arpanet\_graph)

radius = nx.radius(arpanet\_graph)

diameter = nx.diameter(arpanet\_graph)

center = nx.center(arpanet\_graph)

distances = dict(nx.shortest\_path\_length(arpanet\_graph))

print("a. The graph is undirected.")

print(f"b. Vertices: {vertices}")

print(f" Edges: {edges}")

print(f"c. Length (Number of edges): {length}")

print("d. Distance between vertices:")

for node, dist in distances.items():

print(f" {node}: {dist}")

print(f"e. Eccentricity: {eccentricity}")

print(f"f. Radius: {radius}")

print(f"g. Diameter: {diameter}")

print(f"h. Central points of the graph: {center}")

plt.figure(figsize=(10, 8))

pos = nx.spring\_layout(arpanet\_graph) nx.draw(

arpanet\_graph,

pos,

with\_labels=True,

node\_color="lightgreen",

node\_size=2500,

font\_size=10,

font\_weight="bold",

edge\_color="blue",

)

plt.title("Arpanet Map Graph", fontsize=14)

plt.show()

Output :

The graph is undirected.

Vertices: ['SDC', 'RAND', 'SRI', 'UCLA', 'UCSB', 'IPTO', 'BBN', 'MIT']

Edges: [('SDC', 'RAND'), ('SDC', 'UCLA'), ('RAND', 'UCLA'), ('RAND', 'BBN'), ('UCLA', 'UCSB'), ('UCSB', 'SRI'), ('SRI', 'IPTO'), ('SRI', 'UCLA'), ('SRI', 'MIT'), ('IPTO', 'BBN'), ('BBN', 'MIT')]

Length (Number of edges): 11

Distance between vertices:

SDC: {'SDC': 0, 'RAND': 1, 'UCLA': 1, 'UCSB': 2, 'SRI': 2, 'IPTO': 3, 'BBN': 2, 'MIT': 3}

RAND: {'RAND': 0, 'SDC': 1, 'UCLA': 1, 'UCSB': 2, 'SRI': 2, 'IPTO': 2, 'BBN': 1, 'MIT': 2}

SRI: {'SRI': 0, 'UCSB': 1, 'UCLA': 1, 'IPTO': 1, 'MIT': 1, 'BBN': 2, 'SDC': 2, 'RAND': 2}

UCLA: {'UCLA': 0, 'SDC': 1, 'RAND': 1, 'UCSB': 1, 'SRI': 1, 'MIT': 2, 'IPTO': 2, 'BBN': 2}

UCSB: {'UCSB': 0, 'UCLA': 1, 'SRI': 1, 'SDC': 2, 'IPTO': 2, 'MIT': 2, 'RAND': 2, 'BBN': 3}

IPTO: {'IPTO': 0, 'SRI': 1, 'BBN': 1, 'MIT': 2, 'UCSB': 2, 'UCLA': 2, 'RAND': 2, 'SDC': 3}

BBN: {'BBN': 0, 'RAND': 1, 'MIT': 1, 'IPTO': 1, 'SRI': 2, 'UCSB': 3, 'UCLA': 2, 'SDC': 2}

MIT: {'MIT': 0, 'BBN': 1, 'SRI': 1, 'IPTO': 2, 'UCSB': 2, 'UCLA': 2, 'RAND': 2, 'SDC': 3}

Eccentricity: {'SDC': 3, 'RAND': 2, 'SRI': 2, 'UCLA': 2, 'UCSB': 3, 'IPTO': 3, 'BBN': 3, 'MIT': 3}

Radius: 2

Diameter: 3

Central points of the graph: ['RAND', 'SRI', 'UCLA']

