



Bansilal Ramnath Agarwal Charitable Trust's  
Vishwakarma Institute of Information Technology

**Department of  
Artificial Intelligence and Data Science**

**Name:** Siddhesh Dilip Khairnar

**Class:** TY

**Division:** B

**Roll No:** 372028

**Semester:** 6<sup>th</sup>

**Academic Year:** 2023-24

**Subject Name & Code:** Data Science & ADUA32202

**Title of Assignment:** Consider the nodes as (A, B, C, D, E, F, G, H). Take one of the nodes as central node (ego), say A. Write a program to create ego network.

**Date of Performance:** 18-02-2024

**Date of Submission:** 26-02-2024

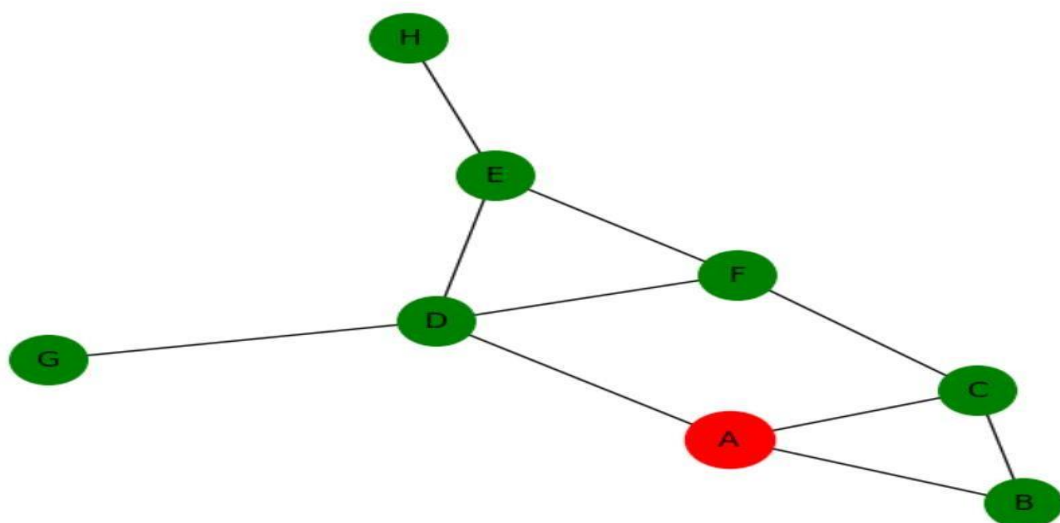
**ASSIGNMENT NO: - 4**

**Aim:** Consider the nodes as (A, B, C, D, E, F, G, H). Take one of the nodes as central node (ego), say A. Write a program to create ego network.

```
#network x for graph generation
import networkx as nx
import matplotlib.pyplot as plt
```

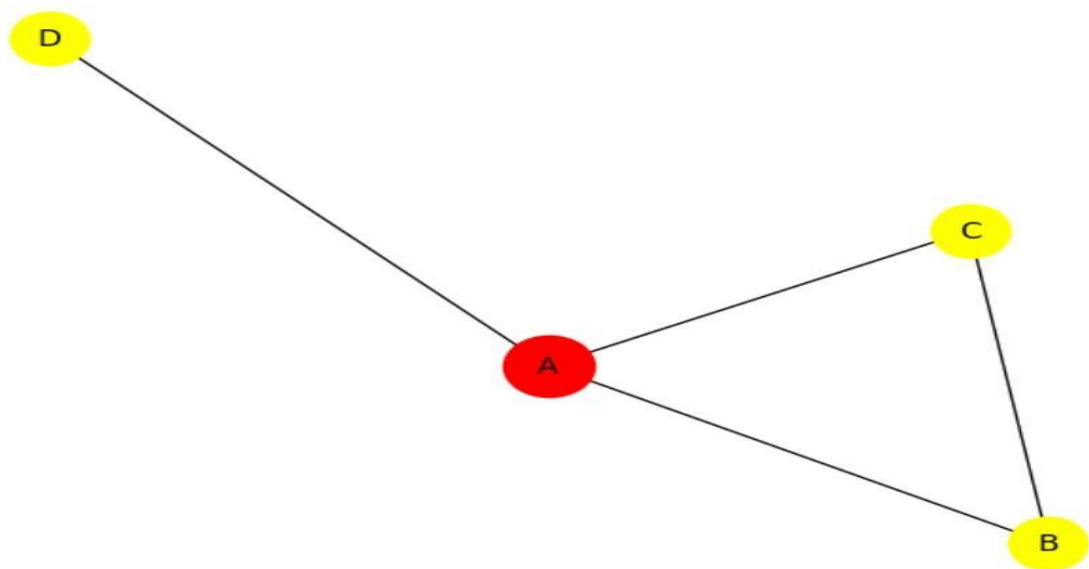
```
#generation of a sample graph
G = nx.Graph()
G.add_edges_from([('A','B'), ('A','C'),
                  ('B','C'),('E','F'),
                  ('D','E'),('A','D'),
                  ('D','G'),('C','F'),
                  ('D','F'),('E','H'),])
```

```
#defining ego as large and red while alters are in blue
#let 'A' be the ego
ego = 'A'
pos = nx.spring_layout(G)
nx.draw(G, pos, node_color="green", node_size=900, with_labels=True)
options = {"node_size": 1200, "node_color": "r"}
nx.draw_networkx_nodes(G, pos, nodelist=[ego], **options)
plt.show()
```



```
#create ego network
hub_ego = nx.ego_graph(G, ego)

#showing the ego network
nx.draw(hub_ego, pos, node_color="yellow", node_size=900, with_labels=True)
nx.draw_networkx_nodes(hub_ego, pos, nodelist= [ego], **options)
plt.show()
```



```
import networkx as nx
import matplotlib.pyplot as plt

# Define the nodes
nodes = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H']

# Create an empty graph
G = nx.Graph()

# Add nodes to the graph
G.add_nodes_from(nodes)

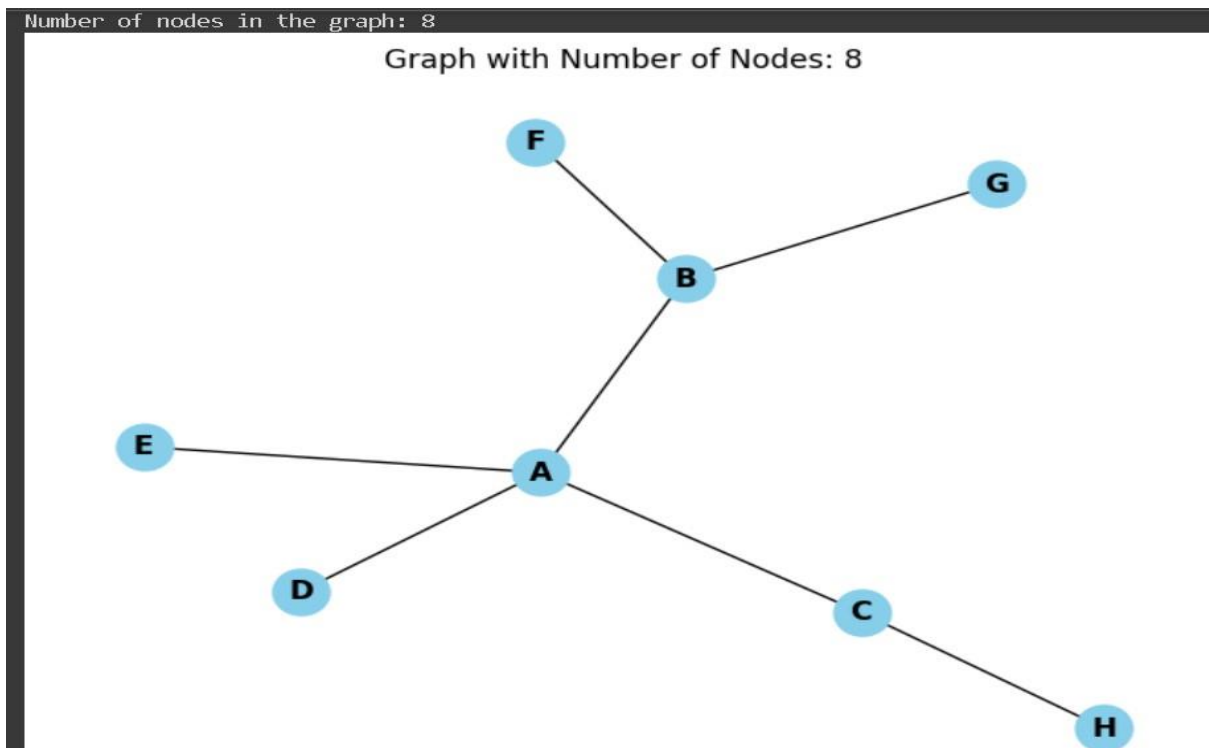
# Define edges (you can modify this based on your network)
edges = [('A', 'B'), ('A', 'C'), ('A', 'D'), ('A', 'E'), ('B', 'F'), ('B', 'G'), ('C', 'H')]

# Add edges to the graph
G.add_edges_from(edges)

# Function to get the number of nodes
def get_number_of_nodes(graph):
    return graph.number_of_nodes()

# Get the number of nodes in the graph
num_nodes = get_number_of_nodes(G)
print("Number of nodes in the graph:", num_nodes)
```

```
# Plot the graph
nx.draw(G, with_labels=True, node_color='skyblue', node_size=500, font_size=12, font_weight='bold')
plt.title("Graph with Number of Nodes: " + str(num_nodes))
plt.show()
```



**Conclusion:** Thus, we have successfully implemented the program for ego network considering the nodes as (A, B, C, D, E, F, G, H).