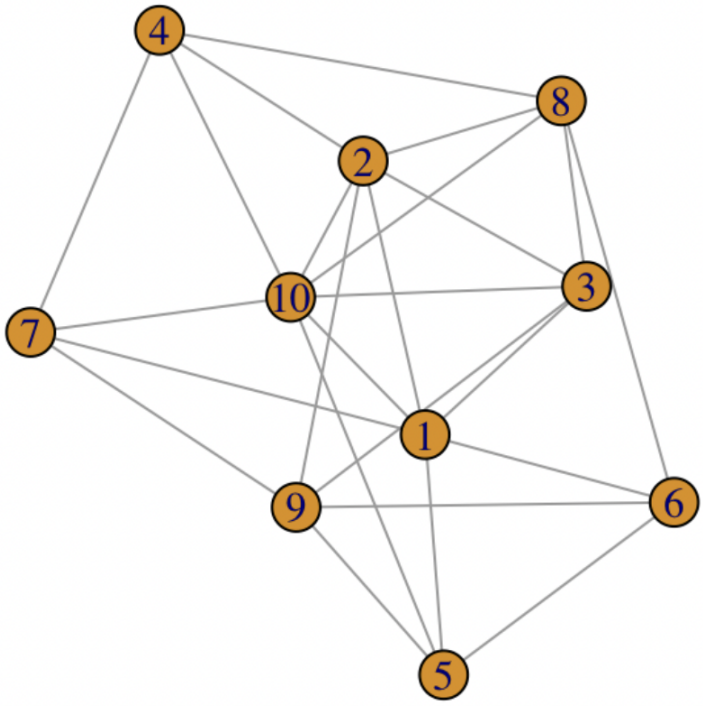
Programme	:	**B.Tech** Semester : **Win Sem 21-22**
Course	:	**Web Mining Lab** Code : **CSE3024**
Faculty	:	**Dr.Bhuvaneswari A** Slot : **L7+L8**
Date	:	**14-03--2022** Marks : **10 Points**

Vaibhav Agarwal

19BCE1413

Exercise 8: **CENTRALITY METRICS – SOCIAL NETWORK ANALYSIS**

Consider the following Facebook friendship network (mutual connection)



1. Find the degree centrality of all nodes
2. Find the neighbors of node 2.
3. Find the average degree of graph
4. Find the density of the graph
5. Find the closeness centrality of Node 10

6. Find all the paths to reach 4 from 6
7. Find the longest shortest path between any two nodes
8. Find the shortest path between any two nodes
9. Find the Betweenness centrality of Node 1.
10. Find the person who has maximum number of connections (friends)

```
G = nx.Graph()
#create a graph
G.add_edge('1','2',relation="neighbour")
G.add_edge('1','3',relation="neighbour")
G.add_edge('1','5',relation="neighbour")
G.add_edge('1','6',relation="neighbour")
G.add_edge('1','10',relation="neighbour")
G.add_edge('2','1',relation="neighbour")
G.add_edge('2','3',relation="neighbour")
G.add_edge('2','4',relation="neighbour")
G.add_edge('2','8',relation="neighbour")
G.add_edge('2','9',relation="neighbour")
G.add_edge('2','10',relation="neighbour")
G.add_edge('3','1',relation="neighbour")
G.add_edge('3','2',relation="neighbour")
G.add_edge('3','8',relation="neighbour")
G.add_edge('3','9',relation="neighbour")
G.add_edge('4','2',relation="neighbour")
G.add_edge('4','7',relation="neighbour")
G.add_edge('4','8',relation="neighbour")
G.add_edge('4','10',relation="neighbour")
G.add_edge('5','1',relation="neighbour")
G.add_edge('5','6',relation="neighbour")
G.add_edge('5','9',relation="neighbour")
G.add_edge('5','10',relation="neighbour")
G.add_edge('6','1',relation="neighbour")
G.add_edge('6','5',relation="neighbour")
G.add_edge('6','8',relation="neighbour")
G.add_edge('6','9',relation="neighbour")
G.add_edge('7','1',relation="neighbour")
G.add_edge('7','4',relation="neighbour")
G.add_edge('7','9',relation="neighbour")
G.add_edge('7','10',relation="neighbour")
G.add_edge('8','2',relation="neighbour")
G.add_edge('8','3',relation="neighbour")
G.add_edge('8','4',relation="neighbour")
G.add_edge('8','6',relation="neighbour")
G.add_edge('9','2',relation="neighbour")
G.add_edge('9','3',relation="neighbour")
G.add_edge('9','5',relation="neighbour")
G.add_edge('9','6',relation="neighbour")
G.add_edge('9','7',relation="neighbour")
G.add_edge('10','1',relation="neighbour")
```

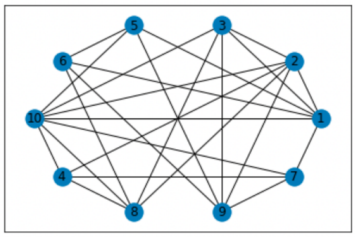
```
G.add_edge('10','3',relation="neighbour")
G.add_edge('10','5',relation="neighbour")
G.add_edge('10','8',relation="neighbour")
G.add_edge('10','4',relation="neighbour")
G.add_edge('10','2',relation="neighbour")
G.add_edge('10','7',relation="neighbour")
G.edges(data=True)
```

Python

```
EdgeDataView([('1', '2', {'relation': 'neighbour'}), ('1', '3', {'relation': 'neighbour'}), ('1', '5', {'relation': 'neighbour'}), ('1', '6', {'relation': 'neighbour'}), ('1', '10', {'relation': 'neighbour'}), ('1', '7', {'relation': 'neighbour'}), ('2', '3', {'relation': 'neighbour'}), ('2', '4', {'relation': 'neighbour'}), ('2', '8', {'relation': 'neighbour'}), ('2', '9', {'relation': 'neighbour'}), ('2', '10', {'relation': 'neighbour'}), ('3', '8', {'relation': 'neighbour'}), ('3', '9', {'relation': 'neighbour'}), ('3', '10', {'relation': 'neighbour'}), ('5', '6', {'relation': 'neighbour'}), ('5', '9', {'relation': 'neighbour'}), ('5', '10', {'relation': 'neighbour'}), ('6', '8', {'relation': 'neighbour'}), ('6', '9', {'relation': 'neighbour'}), ('10', '4', {'relation': 'neighbour'}), ('10', '7', {'relation': 'neighbour'}), ('10', '8', {'relation': 'neighbour'}), ('4', '7', {'relation': 'neighbour'}), ('4', '8', {'relation': 'neighbour'}), ('9', '7', {'relation': 'neighbour'})])
```

```
nx.draw_networkx(G, pos=nx.circular_layout(G),with_labels=True)
plt.show()
```

Python



1) Find the degree centrality of all the nodes

```
degree_centrality = nx.degree_centrality(G)
degree_centrality
```

Python

```
{'1': 0.6666666666666666,
 '10': 0.7777777777777777,
 '2': 0.6666666666666666,
 '3': 0.5555555555555556,
 '4': 0.4444444444444444,
 '5': 0.4444444444444444,
 '6': 0.4444444444444444,
 '7': 0.4444444444444444,
 '8': 0.5555555555555556,
 '9': 0.5555555555555556}
```

2) Find the neighbors of node 2

```
print("Number of neighbors of node 2:")
G['2']
```

Python

```
Number of neighbors of node 2:

AtlasView({'1': {'relation': 'neighbour'}, '3': {'relation': 'neighbour'}, '4': {'relation': 'neighbour'}, '8': {'relation': 'neighbour'}, '9': {'relation': 'neighbour'}, '10':
{'relation': 'neighbour'}})
```

+ Code + Markdown

3) Find the average degree of graph

```
print("Average degree of graph is:")
2*G.number_of_edges() / float(num_of_vertices)
```

Python

```
Average degree of graph is:

5.0
```

4) Find the density of the graph

```
print("Density of graph is:")
nx.density(G)
```

Python

```
Density of graph is:

0.5555555555555556
```

+ Code + Markdown

5) Find the closeness centrality of node 10

```
print("Closeness centrality of node 10 is:");
closeness_centrality = nx.closeness_centrality(G)
closeness_centrality['10']
```

Python

```
Closeness centrality of node 10 is:

0.8181818181818182
```

6) Find all the paths to reach 4 from 6

```
print("All paths from node 4 to 6 are:");
path=nx.all_simple_paths(G,source='4',target='6')
print(list(path))
```

Python

```
All paths from node 4 to 6 are:
[['4', '2', '1', '3', '8', '6'], ['4', '2', '1', '3', '8', '10', '5', '6'], ['4', '2', '1', '3', '8', '10', '5', '9', '6'], ['4', '2', '1', '3', '8', '10', '7', '9', '5', '6'], ['4',
'2', '1', '3', '8', '10', '7', '9', '6'], ['4', '2', '1', '3', '9', '5', '6'], ['4', '2', '1', '3', '9', '5', '10', '8', '6'], ['4', '2', '1', '3', '9', '6'], ['4', '2', '1', '3',
'9', '7', '10', '5', '6'], ['4', '2', '1', '3', '9', '7', '10', '8', '6'], ['4', '2', '1', '3', '10', '5', '6'], ['4', '2', '1', '3', '10', '5', '9', '6'], ['4', '2', '1', '3', '10',
'7', '9', '5', '6'], ['4', '2', '1', '3', '10', '7', '9', '6'], ['4', '2', '1', '3', '10', '8', '6'], ['4', '2', '1', '5', '6'], ['4', '2', '1', '5', '9', '3', '8', '6'], ['4', '2',
'1', '5', '9', '3', '10', '8', '6'], ['4', '2', '1', '5', '9', '6'], ['4', '2', '1', '5', '9', '7', '10', '3', '8', '6'], ['4', '2', '1', '5', '9', '7', '10', '8', '6'], ['4', '2',
'1', '5', '10', '3', '8', '6'], ['4', '2', '1', '5', '10', '3', '9', '6'], ['4', '2', '1', '5', '10', '7', '9', '3', '8', '6'], ['4', '2', '1', '5', '10', '7', '9', '6'], ['4', '2',
'1', '5', '10', '8', '3', '9', '6'], ['4', '2', '1', '5', '10', '8', '6'], ['4', '2', '1', '6'], ['4', '2', '1', '7', '9', '3', '8', '6'], ['4', '2', '1', '7', '9', '3', '8', '10',
'5', '6'], ['4', '2', '1', '7', '9', '3', '10', '5', '6'], ['4', '2', '1', '7', '9', '3', '10', '8', '6'], ['4', '2', '1', '7', '9', '5', '6'], ['4', '2', '1', '7', '9', '5', '10',
'3', '8', '6'], ['4', '2', '1', '7', '9', '5', '10', '8', '6'], ['4', '2', '1', '7', '9', '6'], ['4', '2', '1', '7', '10', '3', '8', '6'], ['4', '2', '1', '7', '10', '3', '9', '5',
'6'], ['4', '2', '1', '7', '10', '3', '9', '6'], ['4', '2', '1', '7', '10', '5', '6'], ['4', '2', '1', '7', '10', '5', '9', '3', '8', '6'], ['4', '2', '1', '7', '10', '5', '9', '6'],
['4', '2', '1', '7', '10', '8', '3', '9', '5', '6'], ['4', '2', '1', '7', '10', '8', '3', '9', '6'], ['4', '2', '1', '7', '10', '8', '6'], ['4', '2', '1', '10', '3', '8', '6'], ['4',
'2', '1', '10', '3', '9', '5', '6'], ['4', '2', '1', '10', '3', '9', '6'], ['4', '2', '1', '10', '5', '6'], ['4', '2', '1', '10', '5', '9', '3', '8', '6'], ['4', '2', '1', '10', '5',
'9', '6'], ['4', '2', '1', '10', '7', '9', '3', '8', '6'], ['4', '2', '1', '10', '7', '9', '5', '6'], ['4', '2', '1', '10', '7', '9', '6'], ['4', '2', '1', '10', '8', '3', '9', '5',
'6'], ['4', '2', '1', '10', '8', '3', '9', '6'], ['4', '2', '1', '10', '8', '6'], ['4', '2', '3', '1', '5', '6'], ['4', '2', '3', '1', '5', '9', '6'], ['4', '2', '3', '1', '5', '9',
'7', '10', '8', '6'], ['4', '2', '3', '1', '5', '10', '7', '9', '6'], ['4', '2', '3', '1', '5', '10', '8', '6'], ['4', '2', '3', '1', '6'], ['4', '2', '3', '1', '7', '9', '5', '6'],
['4', '2', '3', '1', '7', '9', '5', '10', '8', '6'], ['4', '2', '3', '1', '7', '9', '6'], ['4', '2', '3', '1', '7', '10', '5', '6'], ['4', '2', '3', '1', '7', '10', '5', '9', '6'],
['4', '2', '3', '1', '7', '10', '8', '6'], ['4', '2', '3', '1', '10', '5', '6'], ['4', '2', '3', '1', '10', '5', '9', '6'], ['4', '2', '3', '1', '10', '7', '9', '5', '6'], ['4', '2',
'3', '1', '10', '7', '9', '6'], ['4', '2', '3', '1', '10', '8', '6'], ['4', '2', '3', '8', '6'], ['4', '2', '3', '8', '10', '1', '5', '6'], ['4', '2', '3', '8', '10', '1', '5', '9',
'6'], ['4', '2', '3', '8', '10', '1', '6'], ['4', '2', '3', '8', '10', '1', '7', '9', '5', '6'], ['4', '2', '3', '8', '10', '1', '7', '9', '6'], ['4', '2', '3', '8', '10', '5', '1',
'6'], ['4', '2', '3', '8', '10', '5', '1', '7', '9', '6'], ['4', '2', '3', '8', '10', '5', '6'], ['4', '2', '3', '8', '10', '5', '9', '6'], ['4', '2', '3', '8', '10', '5', '9', '7',
'1', '6'], ['4', '2', '3', '8', '10', '7', '1', '5', '6'], ['4', '2', '3', '8', '10', '7', '1', '5', '9', '6'], ['4', '2', '3', '8', '10', '7', '1', '6'], ['4', '2', '3', '8', '10',
'7', '9', '5', '1', '6'], ['4', '2', '3', '8', '10', '7', '9', '5', '6'], ['4', '2', '3', '8', '10', '7', '9', '6'], ['4', '2', '3', '9', '5', '1', '6'], ['4', '2', '3', '9', '5',
'1', '7', '10', '8', '6'], ['4', '2', '3', '9', '5', '1', '10', '8', '6'], ['4', '2', '3', '9', '5', '10', '1', '6'], ['4', '2', '3', '9', '5', '10',
'7', '1', '6'], ['4', '2', '3', '9', '5', '10', '8', '6'], ['4', '2', '3', '9', '6'], ['4', '2', '3', '9', '7', '1', '5', '6'], ['4', '2', '3', '9', '7', '1', '5', '10', '8', '6'],
['4', '2', '3', '9', '7', '1', '6'], ['4', '2', '3', '9', '7', '1', '10', '5', '6'], ['4', '2', '3', '9', '7', '1', '10', '8', '6'], ['4', '2', '3', '9', '7', '10', '1', '5', '6'],
['4', '2', '3', '9', '7', '10', '1', '6'], ['4', '2', '3', '9', '7', '10', '5', '1', '6'], ['4', '2', '3', '9', '7', '10', '5', '6'], ['4', '2', '3', '9', '7', '10', '8', '6'], ['4',
'2', '3', '10', '1', '5', '6'], ['4', '2', '3', '10', '1', '5', '9', '6'], ['4', '2', '3', '10', '1', '6'], ['4', '2', '3', '10', '1', '7', '9', '5', '6'], ['4', '2', '3', '10', '1',

'3', '10', '7', '9', '5', '1', '6'], ['4', '2', '3', '10', '7', '9', '5', '6'], ['4', '2', '3', '10', '7', '9', '6'], ['4', '2', '3', '10', '8', '6'], ['4', '2', '8', '3', '1', '5',
'6'], ['4', '2', '8', '3', '1', '5', '9', '6'], ['4', '2', '8', '3', '1', '5', '10', '7', '9', '6'], ['4', '2', '8', '3', '1', '6'], ['4', '2', '8', '3', '1', '7', '9', '5', '6'],
['4', '2', '8', '3', '1', '7', '9', '6'], ['4', '2', '8', '3', '1', '7', '10', '5', '6'], ['4', '2', '8', '3', '1', '7', '10', '5', '9', '6'], ['4', '2', '8', '3', '1', '10', '5',
'6'], ['4', '2', '8', '3', '1', '10', '5', '9', '6'], ['4', '2', '8', '3', '1', '10', '7', '9', '5', '6'], ['4', '2', '8', '3', '1', '10', '7', '9', '6'], ['4', '2', '8', '3', '9',
'5', '1', '6'], ['4', '2', '8', '3', '9', '5', '6'], ['4', '2', '8', '3', '9', '5', '10', '1', '6'], ['4', '2', '8', '3', '9', '5', '10', '7', '1', '6'], ['4', '2', '8', '3', '9',
'6'], ['4', '2', '8', '3', '9', '7', '1', '5', '6'], ['4', '2', '8', '3', '9', '7', '1', '6'], ['4', '2', '8', '3', '9', '7', '1', '10', '5', '6'], ['4', '2', '8', '3', '9', '7',
'10', '1', '5', '6'], ['4', '2', '8', '3', '9', '7', '10', '1', '6'], ['4', '2', '8', '3', '9', '7', '10', '5', '1', '6'], ['4', '2', '8', '3', '9', '7', '10', '5', '6'], ['4', '2',
'8', '3', '10', '1', '5', '6'], ['4', '2', '8', '3', '10', '1', '5', '9', '6'], ['4', '2', '8', '3', '10', '1', '6'], ['4', '2', '8', '3', '10', '1', '7', '9', '5', '6'], ['4', '2',
'8', '3', '10', '1', '7', '9', '6'], ['4', '2', '8', '3', '10', '5', '1', '6'], ['4', '2', '8', '3', '10', '5', '1', '7', '9', '6'], ['4', '2', '8', '3', '10', '5', '6'], ['4', '2',
'8', '3', '10', '5', '9', '6'], ['4', '2', '8', '3', '10', '5', '9', '7', '1', '6'], ['4', '2', '8', '3', '10', '7', '1', '5', '6'], ['4', '2', '8', '3', '10', '7', '1', '5', '9',
'6'], ['4', '2', '8', '3', '10', '7', '1', '6'], ['4', '2', '8', '3', '10', '7', '9', '5', '1', '6'], ['4', '2', '8', '3', '10', '7', '9', '5', '6'], ['4', '2', '8', '3', '10', '7',
'9', '6'], ['4', '2', '8', '6'], ['4', '2', '8', '10', '1', '3', '9', '5', '6'], ['4', '2', '8', '10', '1', '3', '9', '6'], ['4', '2', '8', '10', '1', '5', '6'], ['4', '2', '8',
'10', '1', '5', '9', '6'], ['4', '2', '8', '10', '1', '6'], ['4', '2', '8', '10', '1', '7', '9', '5', '6'], ['4', '2', '8', '10', '1', '7', '9', '6'], ['4', '2', '8', '10', '3', '1',
'5', '6'], ['4', '2', '8', '10', '3', '1', '5', '9', '6'], ['4', '2', '8', '10', '3', '1', '6'], ['4', '2', '8', '10', '3', '1', '7', '9', '5', '6'], ['4', '2', '8', '10', '3', '1',
'7', '9', '6'], ['4', '2', '8', '10', '3', '9', '5', '1', '6'], ['4', '2', '8', '10', '3', '9', '5', '6'], ['4', '2', '8', '10', '3', '9', '6'], ['4', '2', '8', '10', '3', '9', '7',
'1', '5', '6'], ['4', '2', '8', '10', '3', '9', '7', '1', '6'], ['4', '2', '8', '10', '5', '1', '3', '9', '6'], ['4', '2', '8', '10', '5', '1', '6'], ['4', '2', '8', '10', '5', '1',
'7', '9', '6'], ['4', '2', '8', '10', '5', '6'], ['4', '2', '8', '10', '5', '9', '3', '1', '6'], ['4', '2', '8', '10', '5', '9', '6'], ['4', '2', '8', '10', '7', '1', '5', '6'], ['4',
'2', '8', '10', '7', '1', '3', '9', '5', '6'], ['4', '2', '8', '10', '7', '1', '3', '9', '6'], ['4', '2', '8', '10', '7', '1', '5', '6'], ['4', '2', '8', '10', '7', '1', '5', '9',
'6'], ['4', '2', '8', '10', '7', '1', '6'], ['4', '2', '8', '10', '7', '9', '3', '1', '5', '6'], ['4', '2', '8', '10', '7', '9', '5', '1', '6'], ['4', '2', '8', '10', '7', '9', '5',
'6'], ['4', '2', '8', '10', '7', '9', '6'], ['4', '2', '9', '3', '1', '5', '6'], ['4', '2', '9', '3', '1', '5', '10', '8', '6'], ['4', '2', '9', '3', '1', '6'], ['4', '2', '9', '3',
'1', '7', '10', '5', '6'], ['4', '2', '9', '3', '1', '7', '10', '8', '6'], ['4', '2', '9', '3', '1', '10', '5', '6'], ['4', '2', '9', '3', '1', '10', '8', '6'], ['4', '2', '9', '3',
'8', '6'], ['4', '2', '9', '3', '8', '10', '1', '5', '6'], ['4', '2', '9', '3', '8', '10', '1', '6'], ['4', '2', '9', '3', '8', '10', '5', '1', '6'], ['4', '2', '9', '3', '8', '10',
'5', '6'], ['4', '2', '9', '3', '8', '10', '7', '1', '5', '6'], ['4', '2', '9', '3', '8', '10', '7', '1', '6'], ['4', '2', '9', '3', '8', '10', '7', '5', '1', '6'], ['4', '2', '9',
'3', '10', '1', '5', '6'], ['4', '2', '9', '3', '10', '1', '5', '9', '6'], ['4', '2', '9', '3', '10', '1', '6'], ['4', '2', '9', '3', '10', '5', '1', '6'], ['4', '2', '9', '3', '10', '5', '6'], ['4', '2', '9', '3', '10', '7', '1', '5', '6'], ['4', '2', '9', '3', '10', '7', '1', '6'], ['4', '2', '9', '3', '10', '8', '6'], ['4', '2', '9', '5', '1', '3', '8', '6'], ['4', '2', '9', '5', '1', '3', '10', '8', '6'],
['4', '2', '9', '5', '1', '6'], ['4', '2', '9', '5', '1', '7', '10', '3', '8', '6'], ['4', '2', '9', '5', '1', '7', '10', '8', '6'], ['4', '2', '9', '5', '1', '10', '3', '8', '6'],
['4', '2', '9', '5', '1', '10', '8', '6'], ['4', '2', '9', '5', '6'], ['4', '2', '9', '5', '10', '1', '3', '8', '6'], ['4', '2', '9', '5', '10', '1', '6'], ['4', '2', '9', '5', '10',
'3', '1', '6'], ['4', '2', '9', '5', '10', '3', '8', '6'], ['4', '2', '9', '5', '10', '7', '1', '3', '8', '6'], ['4', '2', '9', '5', '10', '7', '1', '6'], ['4', '2', '9', '5', '10',
'8', '3', '1', '6'], ['4', '2', '9', '5', '10', '8', '6'], ['4', '2', '9', '6'], ['4', '2', '9', '7', '1', '3', '8', '6'], ['4', '2', '9', '7', '1', '5', '6'], ['4', '2', '9', '7', '1', '5', '10', '3', '8', '6'],
['4', '2', '9', '7', '1', '5', '10', '8', '6'], ['4', '2', '9', '7', '1', '6'], ['4', '2', '9', '7', '1', '10', '3', '8', '6'], ['4', '2', '9', '7', '1', '10', '5', '6'], ['4', '2',
'9', '7', '1', '10', '8', '6'], ['4', '2', '9', '7', '10', '1', '3', '8', '6'], ['4', '2', '9', '7', '10', '1', '5', '6'], ['4', '2', '9', '7',
'10', '3', '1', '5', '6'], ['4', '2', '9', '7', '10', '3', '1', '6'], ['4', '2', '9', '7', '10', '3', '8', '6'], ['4', '2', '9', '7', '10', '5', '1', '3', '8', '6'], ['4', '2', '9',
'7', '10', '5', '1', '6'], ['4', '2', '9', '7', '10', '5', '6'], ['4', '2', '9', '7', '10', '8', '3', '1', '5', '6'], ['4', '2', '9', '7', '10', '8', '3', '1', '6'], ['4', '2', '9',
```

'2', '10', '1', '5', '9', '3', '8', '6'], ['4', '2', '10', '1', '5', '9', '6'], ['4', '2', '10', '1', '6'], ['4', '2', '10', '1', '7', '9', '3', '8', '6'], ['4', '2', '10', '1', '7',
'9', '5', '6'], ['4', '2', '10', '1', '7', '9', '6'], ['4', '2', '10', '3', '1', '5', '6'], ['4', '2', '10', '3', '1', '5', '9', '6'], ['4', '2', '10', '3', '1', '6'], ['4', '2',
'10', '3', '1', '7', '9', '5', '6'], ['4', '2', '10', '3', '1', '7', '9', '6'], ['4', '2', '10', '3', '8', '6'], ['4', '2', '10', '3', '9', '5', '1', '6'], ['4', '2', '10', '3', '9',
'5', '6'], ['4', '2', '10', '3', '9', '6'], ['4', '2', '10', '3', '9', '7', '1', '5', '6'], ['4', '2', '10', '3', '9', '7', '1', '6'], ['4', '2', '10', '5', '1', '3', '8', '6'],
['4', '2', '10', '5', '1', '3', '9', '6'], ['4', '2', '10', '5', '1', '6'], ['4', '2', '10', '5', '1', '7', '9', '3', '8', '6'], ['4', '2', '10', '5', '1', '7', '9', '6'], ['4', '2',
'10', '5', '6'], ['4', '2', '10', '5', '9', '3', '1', '6'], ['4', '2', '10', '5', '9', '3', '8', '6'], ['4', '2', '10', '5', '9', '6'], ['4', '2', '10', '5', '9', '7', '1', '3', '8',
'6'], ['4', '2', '10', '5', '9', '7', '1', '6'], ['4', '2', '10', '7', '1', '3', '8', '6'], ['4', '2', '10', '7', '1', '3', '9', '5', '6'], ['4', '2', '10', '7', '1', '3', '9', '6'],
['4', '2', '10', '7', '1', '5', '6'], ['4', '2', '10', '7', '1', '5', '9', '3', '8', '6'], ['4', '2', '10', '7', '1', '5', '9', '6'], ['4', '2', '10', '7', '1', '6'], ['4', '2',
'10', '7', '9', '3', '1', '5', '6'], ['4', '2', '10', '7', '9', '3', '1', '6'], ['4', '2', '10', '7', '9', '3', '8', '6'], ['4', '2', '10', '7', '9', '5', '1', '3', '8', '6'], ['4',
'2', '10', '7', '9', '5', '1', '6'], ['4', '2', '10', '7', '9', '5', '6'], ['4', '2', '10', '7', '9', '6'], ['4', '2', '10', '8', '3', '1', '5', '6'], ['4', '2', '10', '8', '3', '1',
'5', '9', '6'], ['4', '2', '10', '8', '3', '1', '6'], ['4', '2', '10', '8', '3', '1', '7', '9', '5', '6'], ['4', '2', '10', '8', '3', '1', '7', '9', '6'], ['4', '2', '10', '8', '3',
'9', '5', '1', '6'], ['4', '2', '10', '8', '3', '9', '5', '6'], ['4', '2', '10', '8', '3', '9', '6'], ['4', '2', '10', '8', '3', '9', '7', '1', '5', '6'], ['4', '2', '10', '8', '3',
'9', '7', '1', '6'], ['4', '2', '10', '8', '6'], ['4', '7', '1', '2', '3', '8', '6'], ['4', '7', '1', '2', '3', '8', '10', '5', '6'], ['4', '7', '1', '2', '3', '8', '10', '5', '9',
'6'], ['4', '7', '1', '2', '3', '9', '5', '6'], ['4', '7', '1', '2', '3', '9', '5', '10', '8', '6'], ['4', '7', '1', '2', '3', '9', '6'], ['4', '7', '1', '2', '3', '10', '5', '6'],
['4', '7', '1', '2', '3', '10', '5', '9', '6'], ['4', '7', '1', '2', '3', '10', '8', '6'], ['4', '7', '1', '2', '8', '3', '9', '5', '6'], ['4', '7', '1', '2', '8', '3', '9', '6'],
['4', '7', '1', '2', '8', '3', '10', '5', '6'], ['4', '7', '1', '2', '8', '3', '10', '5', '9', '6'], ['4', '7', '1', '2', '8', '6'], ['4', '7', '1', '2', '8', '10', '3', '9', '5',
'6'], ['4', '7', '1', '2', '8', '10', '3', '9', '6'], ['4', '7', '1', '2', '8', '10', '5', '6'], ['4', '7', '1', '2', '8', '10', '5', '9', '6'], ['4', '7', '1', '2', '9', '3', '8',
'6'], ['4', '7', '1', '2', '9', '3', '8', '10', '5', '6'], ['4', '7', '1', '2', '9', '3', '10', '5', '6'], ['4', '7', '1', '2', '9', '3', '10', '8', '6'], ['4', '7', '1', '2', '9',
'5', '6'], ['4', '7', '1', '2', '9', '5', '10', '3', '8', '6'], ['4', '7', '1', '2', '9', '5', '10', '8', '6'], ['4', '7', '1', '2', '9', '6'], ['4', '7', '1', '2', '10', '3', '8',
'6'], ['4', '7', '1', '2', '10', '3', '9', '5', '6'], ['4', '7', '1', '2', '10', '3', '9', '6'], ['4', '7', '1', '2', '10', '5', '6'], ['4', '7', '1', '2', '10', '5', '9', '3', '8',
'6'], ['4', '7', '1', '2', '10', '5', '9', '6'], ['4', '7', '1', '2', '10', '8', '3', '9', '5', '6'], ['4', '7', '1', '2', '10', '8', '3', '9', '6'], ['4', '7', '1', '2', '10', '8',
'6'], ['4', '7', '1', '3', '2', '8', '6'], ['4', '7', '1', '3', '2', '8', '10', '5', '6'], ['4', '7', '1', '3', '2', '8', '10', '5', '9', '6'], ['4', '7', '1', '3', '2', '9', '5',
'6'], ['4', '7', '1', '3', '2', '9', '5', '10', '8', '6'], ['4', '7', '1', '3', '2', '9', '6'], ['4', '7', '1', '3', '2', '10', '5', '6'], ['4', '7', '1', '3', '2', '10', '5', '9',
'6'], ['4', '7', '1', '3', '2', '10', '8', '6'], ['4', '7', '1', '3', '8', '2', '9', '5', '6'], ['4', '7', '1', '3', '8', '2', '9', '6'], ['4', '7', '1', '3', '8', '2', '10', '5',
'6'], ['4', '7', '1', '3', '8', '2', '10', '5', '9', '6'], ['4', '7', '1', '3', '8', '6'], ['4', '7', '1', '3', '8', '10', '2', '9', '5', '6'], ['4', '7', '1', '3', '8', '10', '2',
'9', '6'], ['4', '7', '1', '3', '8', '10', '5', '6'], ['4', '7', '1', '3', '8', '10', '5', '9', '6'], ['4', '7', '1', '3', '9', '2', '8', '6'], ['4', '7', '1', '3', '9', '2', '8',
'10', '5', '6'], ['4', '7', '1', '3', '9', '2', '10', '5', '6'], ['4', '7', '1', '3', '9', '2', '10', '8', '6'], ['4', '7', '1', '3', '9', '5', '6'], ['4', '7', '1', '3', '9', '5',
'10', '2', '8', '6'], ['4', '7', '1', '3', '9', '5', '10', '8', '6'], ['4', '7', '1', '3', '9', '6'], ['4', '7', '1', '3', '10', '2', '8', '6'], ['4', '7', '1', '3', '10', '2', '9',
'5', '6'], ['4', '7', '1', '3', '10', '2', '9', '6'], ['4', '7', '1', '3', '10', '5', '6'], ['4', '7', '1', '3', '10', '5', '9', '2', '8', '6'], ['4', '7', '1', '3', '10', '5', '9',
'6'], ['4', '7', '1', '3', '10', '8', '2', '9', '5', '6'], ['4', '7', '1', '3', '10', '8', '2', '9', '6'], ['4', '7', '1', '3', '10', '8', '6'], ['4', '7', '1', '5', '6'], ['4', '7',
'1', '5', '9', '2', '3', '8', '6'], ['4', '7', '1', '5', '9', '2', '3', '10', '8', '6'], ['4', '7', '1', '5', '9', '2', '8', '6'], ['4', '7', '1', '5', '9', '2', '10', '3', '8',
'6'], ['4', '7', '1', '5', '9', '2', '10', '8', '6'], ['4', '7', '1', '5', '9', '3', '2', '8', '6'], ['4', '7', '1', '5', '9', '3', '2', '10', '8', '6'], ['4', '7', '1', '5', '9',
'3', '8', '6'], ['4', '7', '1', '5', '9', '3', '10', '2', '8', '6'], ['4', '7', '1', '5', '9', '3', '10', '8', '6'], ['4', '7', '1', '5', '9', '6'], ['4', '7', '1', '5', '10', '2',
'3', '8', '6'], ['4', '7', '1', '5', '10', '2', '3', '9', '6'], ['4', '7', '1', '5', '10', '2', '8', '3', '9', '6'], ['4', '7', '1', '5', '10', '2', '8', '6'], ['4', '7', '1', '5',
'10', '2', '9', '3', '8', '6'], ['4', '7', '1', '5', '10', '2', '9', '6'], ['4', '7', '1', '5', '10', '3', '2', '8', '6'], ['4', '7', '1', '5', '10', '3', '2', '9', '6'], ['4', '7',
'1', '5', '10', '3', '8', '2', '9', '6'], ['4', '7', '1', '5', '10', '3', '8', '6'], ['4', '7', '1', '5', '10', '3', '9', '2', '8', '6'], ['4', '7', '1', '5', '10', '3', '9', '6'],

7) Find the longest shortest path between any two nodes

```
def max_length_list(input_list):
    max_length = max(len(x) for x in input_list )
    for i in input_list:
      print(i)
      if len(i) == max_length:
        return(max_length,i)
ShortestPaths=[]
print("Longest Shortest path between any two nodes:");
for i in range(num_of_vertices) :
  for j in range(num_of_vertices):
    ShortestPaths.append(nx.shortest_path(G,source=vertices_list[i],target=vertices_list[j]))
print(max_length_list(ShortestPaths))
```

[ ] Python

```
Longest Shortest path between any two nodes:
['1']
['1', '2']
['1', '3']
['1', '2', '4']
(3, ['1', '2', '4'])
```

8) Find the shortest path between any two nodes

```
print("Shortest path between any two nodes:");
for i in range(num_of_vertices) :
    for j in range(num_of_vertices) :
        print(nx.shortest_path(G,source=vertices_list[i],target=vertices_list[j]))
```

Python

```
Shortest path between any two nodes:
['1']
['1', '2']
['1', '3']
['1', '2', '4']
['1', '5']
['1', '6']
['1', '7']
['1', '2', '8']
['1', '2', '9']
['1', '10']
['2', '1']
['2']
['2', '3']
['2', '4']
['2', '1', '5']
['2', '1', '6']
['2', '4', '7']
['2', '8']
['2', '9']
['2', '10']
['3', '1']
['3', '2']
['3']
['3', '2', '4']
```

show more (open the raw output data in a text editor) ...

```
['10', '1', '6']
['10', '7']
['10', '8']
['10', '2', '9']
['10']
```

9) Find the betweeness centrality of Node 1

```
print("Betweeness centrality of node 1:");
betweenness_centrality=nx.betweenness_centrality_source(G,normalized=True, weight=None, sources=None)
betweenness_centrality['1']
```

Python

```
Betweeness centrality of node 1:

0.08564814814814814
```

10) Find the person who has maximum number of connections

+ Code + Markdown

```
print("Person who has maximum number of connections:");
max(dict(G.degree()).items(), key = lambda x : x[1])
```

Python

```
Person who has maximum number of connections:

('10', 7)
```