

11508 Muzamil Khan SNA Assignment 1

Collaboration Networks (ca-HepTh.txt 9,877 nodes/25,998 Edges)

```
In [45]: import networkx as nx
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [47]: name = ['source', 'target']
df = pd.read_csv("CA-HepTh.txt", sep="\t")
```

```
In [48]: df.head(20)
```

```
Out[48]:
```

	FromNodeId	ToNodeId
0	24325	24394
1	24325	40517
2	24325	58507
3	24394	3737
4	24394	3905
5	24394	7237
6	24394	12715
7	24394	13648
8	24394	13659
9	24394	13664
10	24394	14304
11	24394	14823
12	24394	17370
13	24394	18956
14	24394	19615
15	24394	19660
16	24394	21669
17	24394	23106
18	24394	24325
19	24394	24832

Graph

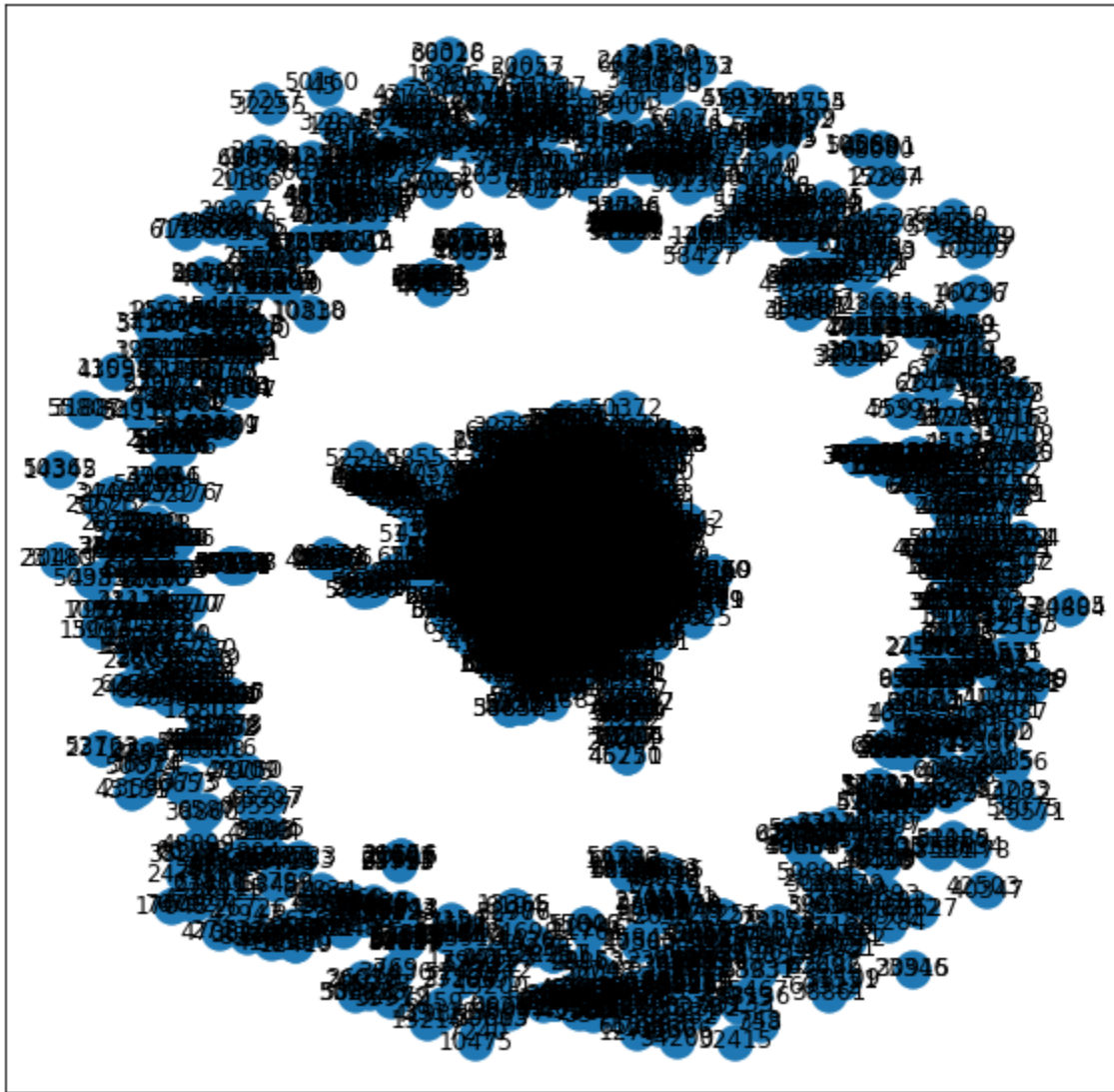
```
In [51]: G = nx.from_pandas_edgelist(df, source='FromNodeId', target='ToNodeId')

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

```

nx.draw_networkx(G)
plt.show()

```



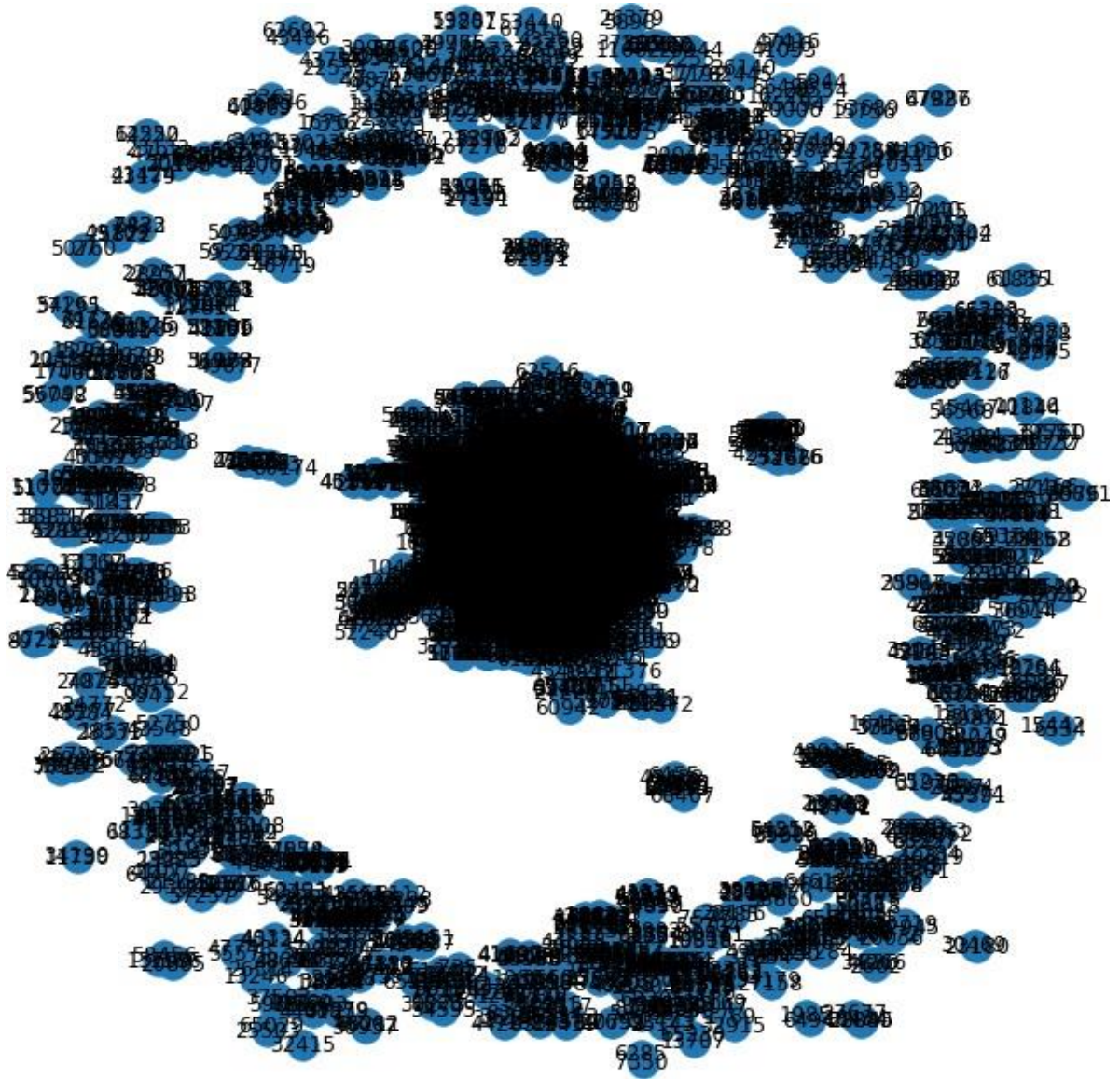
Directed Graph (Di Graph)

```

In [54]: G = nx.from_pandas_edgelist(df, source='FromNodeId', target='ToNodeId', create_using=nx

plt.figure(figsize=(10, 10))
nx.draw(G, with_labels=True)
plt.show()

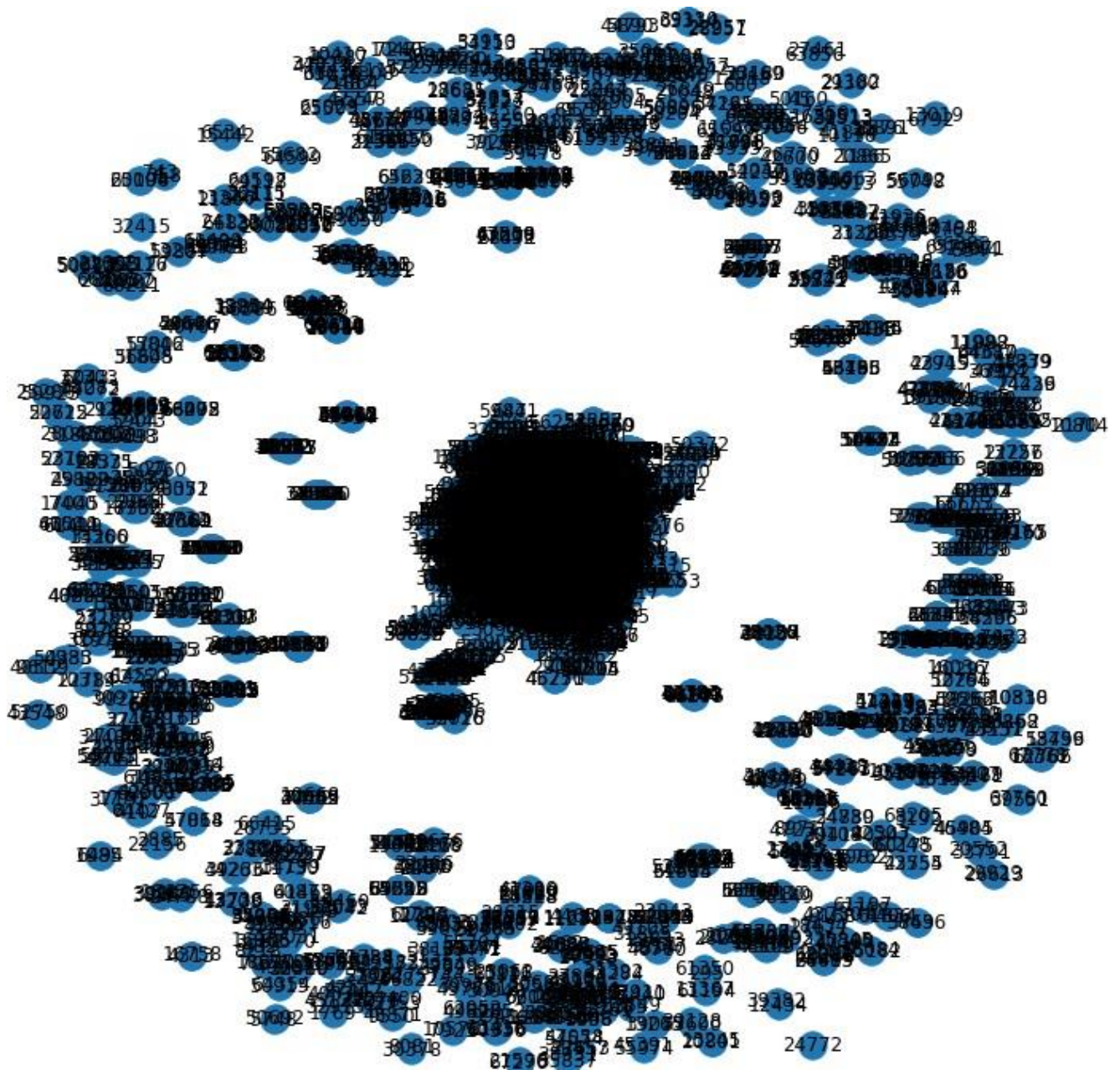
```



Multi Graph

```
In [55]: G = nx.from_pandas_edgelist(df, source='FromNodeId', target='ToNodeId', create_using=nx

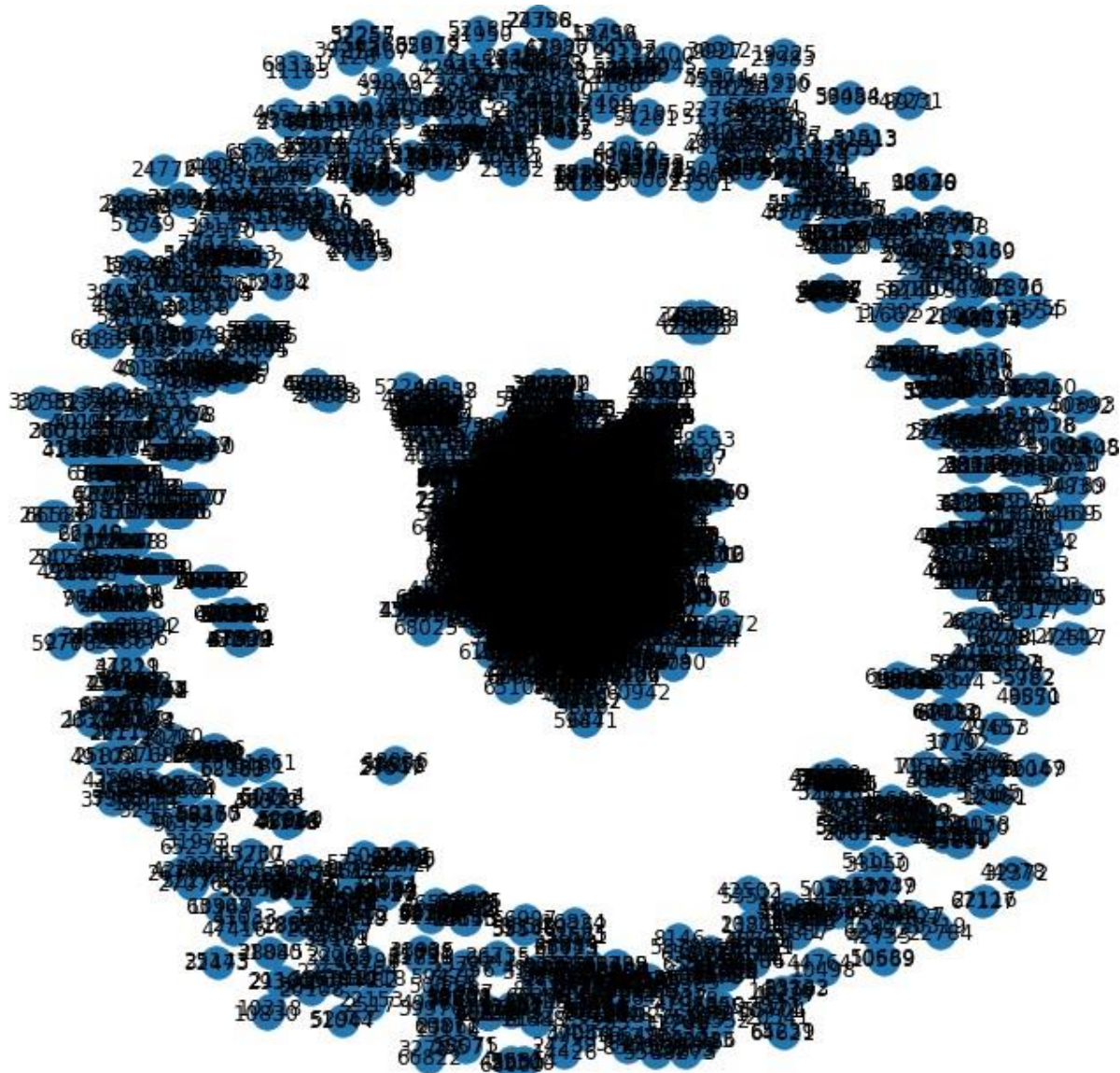
plt.figure(figsize=(10, 10))
nx.draw(G, with_labels=True)
plt.show()
```

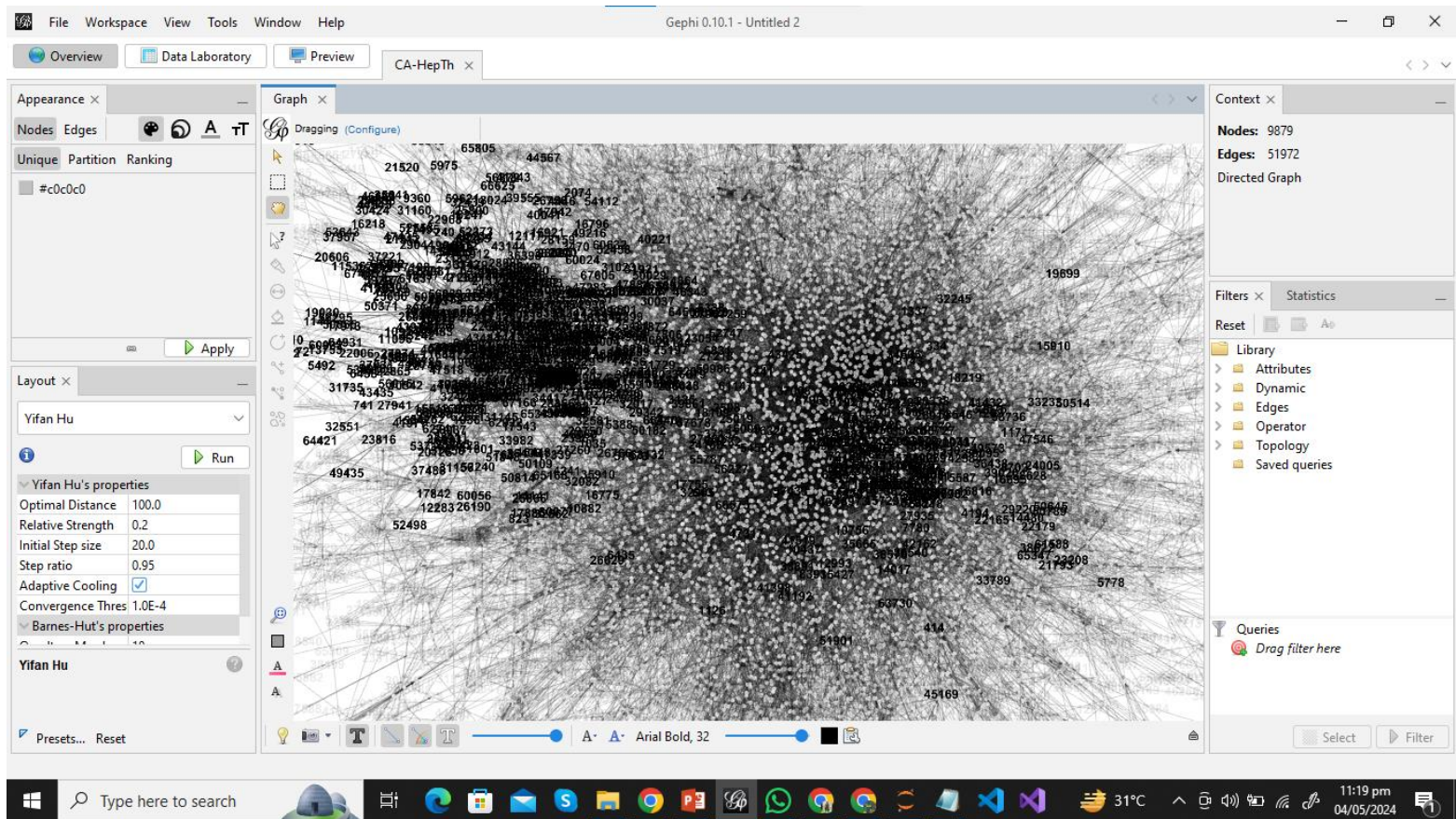
MultiDi Graph

```
In [56]: G = nx.from_pandas_edgelist(df, source='FromNodeId', target='ToNodeId', create_using=nx

plt.figure(figsize=(10, 10))
nx.draw(G, with_labels=True)
plt.show()
```



GIPHE SCREENSHOT:



Dynamic Face-to-Face Interaction Networks (451 nodes/3,126,993 Edges)

```
In [58]: df1 = pd.read_csv("network_list.csv")
```

```
In [59]: df1.head(20)
```

```
Out[59]:
```

	NETWORK	NUMBER_OF_PARTICIPANTS
0	0	7
1	1	8
2	2	7
3	3	7
4	4	8
5	5	8
6	6	7
7	7	6
8	8	7
9	9	8

10	10	7
11	11	7
12	12	8
13	13	7
14	14	8
15	15	6
16	16	6
17	17	7
18	18	8
19	19	5

```
In [61]: us_graph = nx.from_pandas_edgelist(df1, source = 'NETWORK', target = 'NUMBER_OF_PARTICI
us_graph.nodes()
```

```
Out[61]: NodeView((0, 7, 1, 8, 2, 3, 4, 5, 6, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43,
44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61))
```

```
In [63]: len(us_graph.nodes())
```

```
Out[63]: 62
```

```
In [64]: us_graph.edges()
```

```
Out[64]: EdgeView([(0, 7), (7, 2), (7, 3), (7, 6), (7, 8), (7, 10), (7, 11), (7, 13), (7, 17),
(7, 26), (7, 35), (7, 36), (7, 38), (7, 39), (7, 43), (7, 46), (7, 47), (7, 48), (7, 4
9), (7, 51), (7, 55), (7, 59), (7, 60), (1, 8), (8, 4), (8, 5), (8, 9), (8, 12), (8, 1
4), (8, 18), (8, 21), (8, 22), (8, 23), (8, 25), (8, 27), (8, 28), (8, 29), (8, 30), (8,
31), (8, 32), (8, 33), (8, 34), (8, 37), (8, 40), (8, 41), (8, 42), (8, 53), (8, 54),
(8, 56), (8, 57), (8, 58), (8, 61), (5, 19), (5, 24), (6, 15), (6, 16), (6, 20), (6, 4
4), (6, 45), (6, 50), (6, 52)])
```

```
In [65]: len(us_graph.nodes())
```

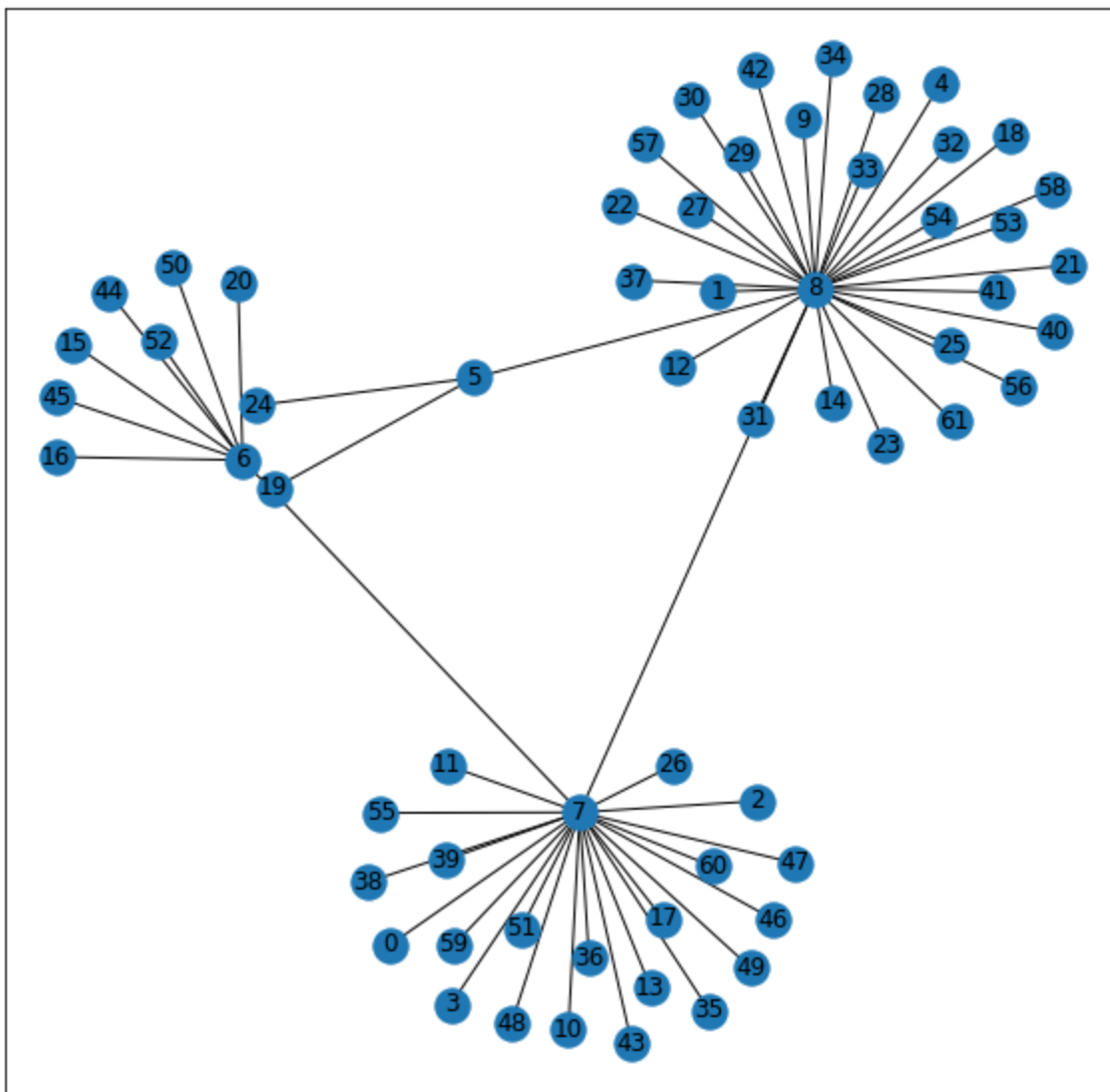
```
Out[65]: 62
```

Graph

```
In [68]: plt.figure(figsize = (10, 10))
```

```
nx.draw_networkx(us_graph)
plt.show
```

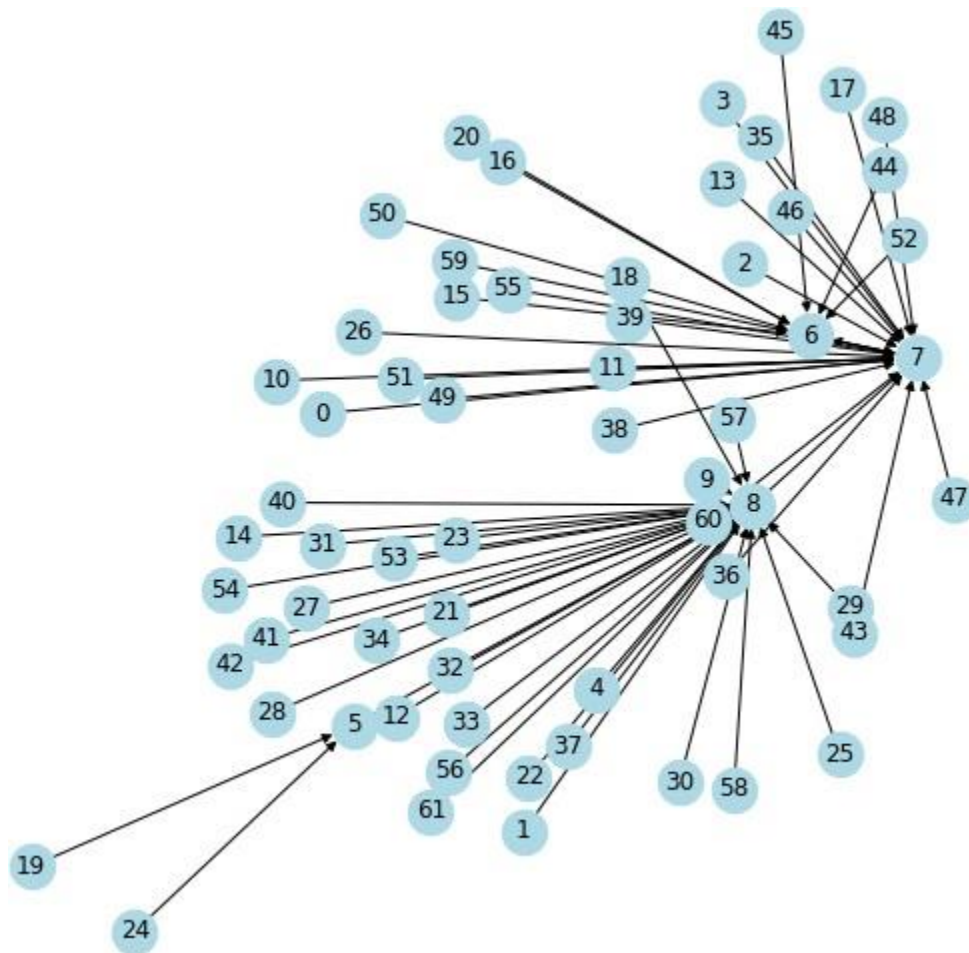
```
Out[68]: <function matplotlib.pyplot.show(close=None, block=None)>
```



Directed Graph (Di Graph)

```
In [70]: G_dir = nx.from_pandas_edgelist(dfl, source = 'NETWORK', target = 'NUMBER_OF_PARTICIPAN

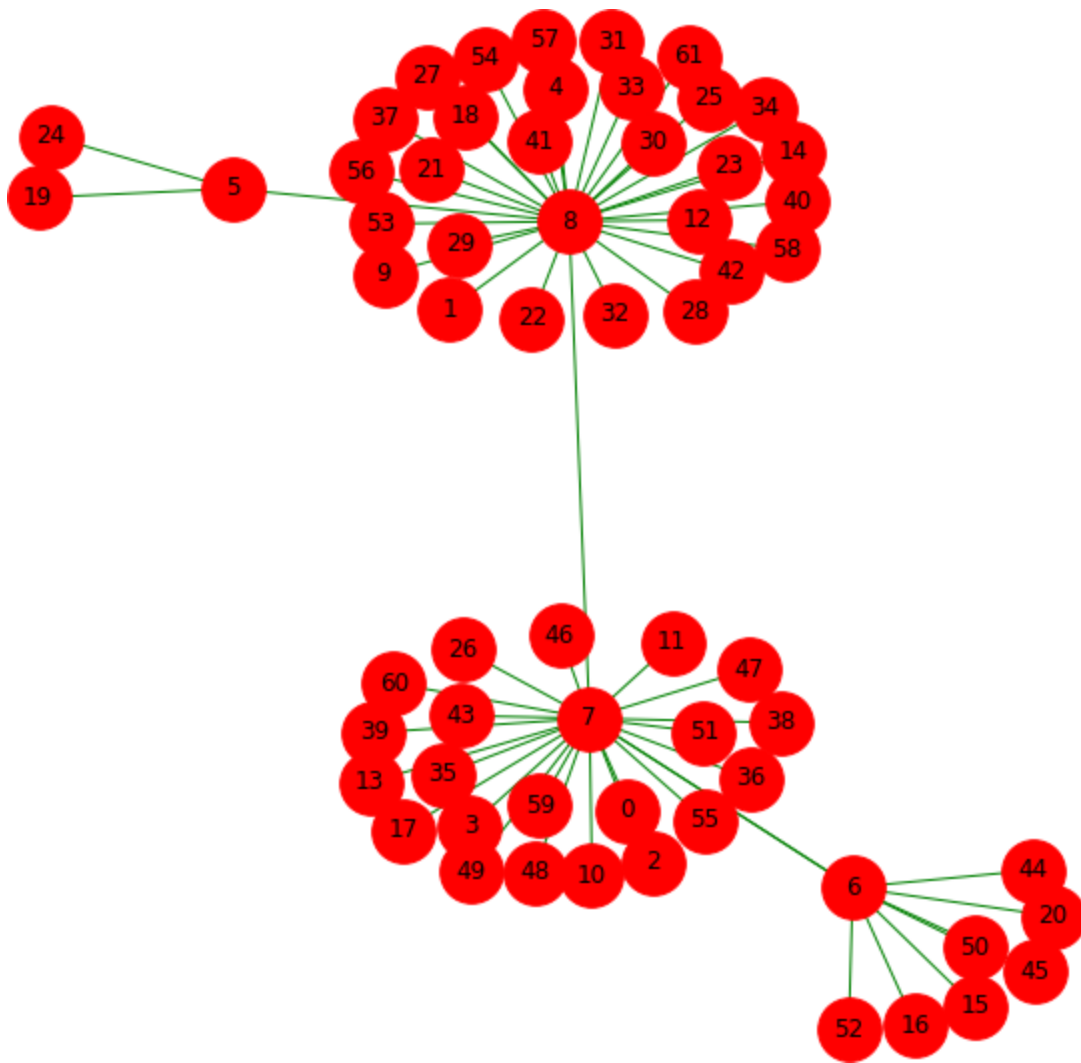
fig = plt.subplots(figsize = (10, 10))
nx.draw(G_dir, with_labels = True, node_color = 'lightblue', node_size = 500, arrowsize
plt.show()
```

Multi Graph

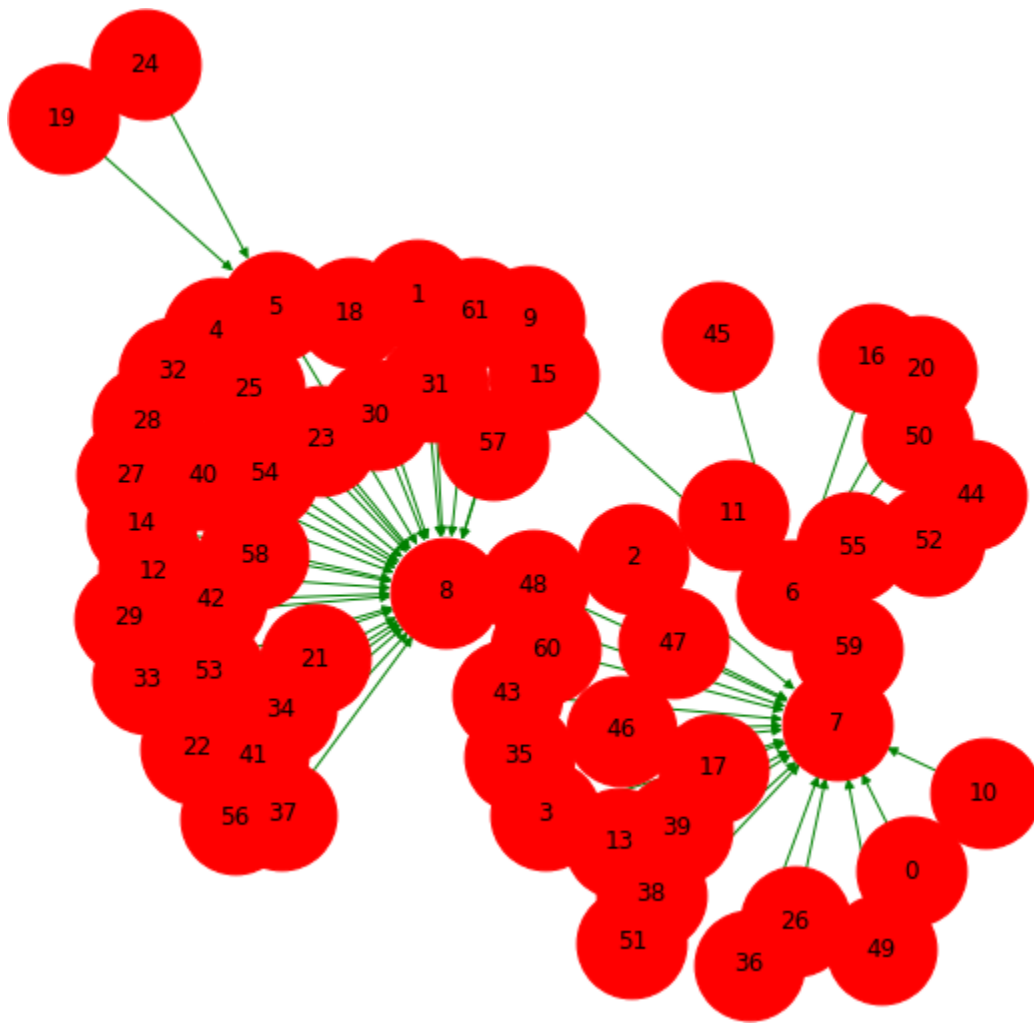
```
In [71]: G_dir = nx.from_pandas_edgelist(df1, source = 'NETWORK', target = 'NUMBER_OF_PARTICIPAN')

fig = plt.subplots(figsize = (10, 10))
nx.draw(G_dir, with_labels = True, node_color = 'r', edge_color = 'g', node_size = 1000)
plt.show()
```

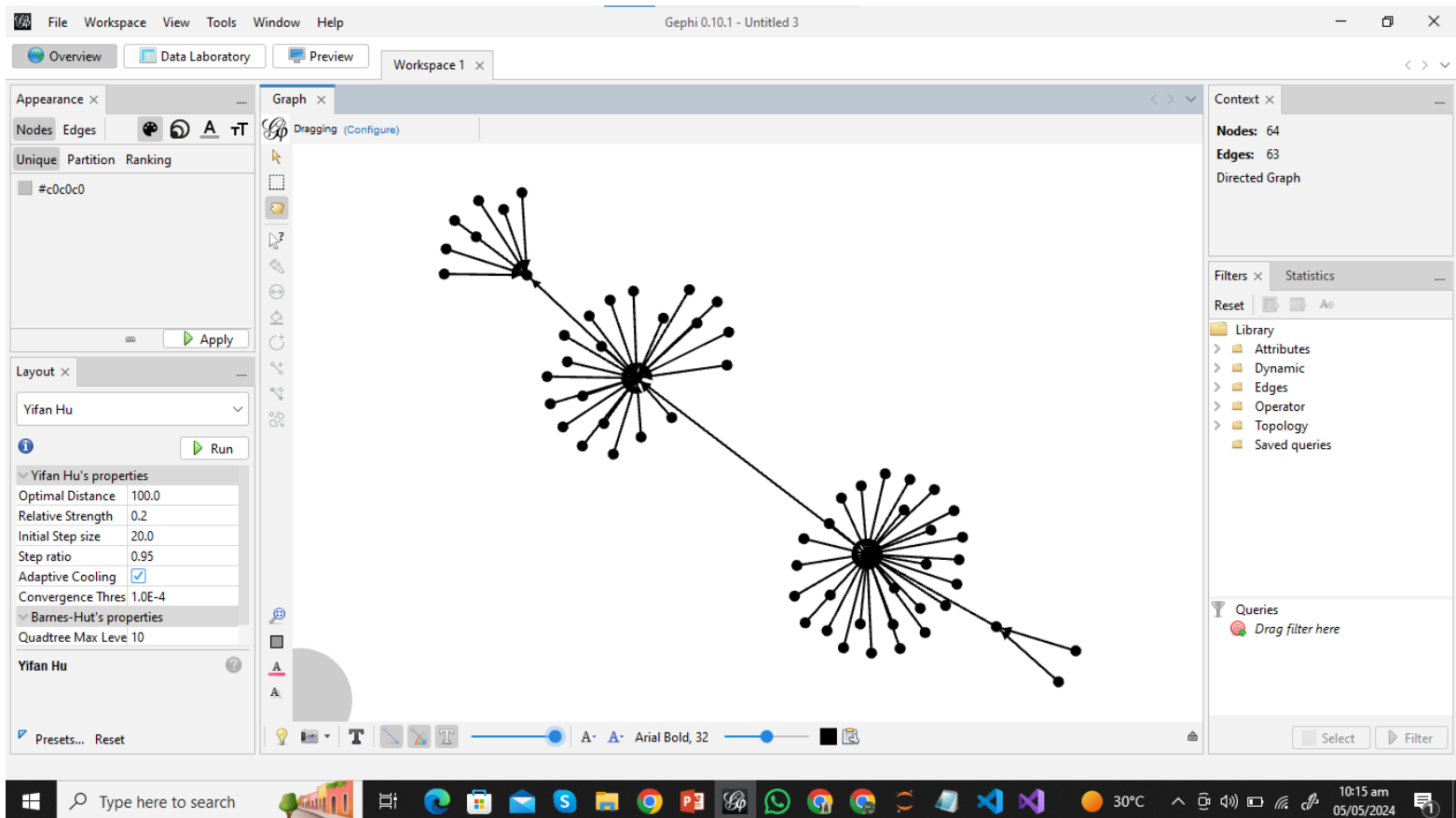


MultiDiGraph

```
In [73]: G_dir = nx.from_pandas_edgelist(df1, source = 'NETWORK', target = 'NUMBER_OF_PARTICIPAN  
  
fig = plt.subplots(figsize = (10, 10))  
nx.draw(G_dir, with_labels = True, node_color = 'r', edge_color = 'g', node_size = 3000  
plt.show()
```



GIPHE SCREENSHOT:



Temporal networks (nodes 24818 / edges 5066550)

```
In [89]: name_row = ['From', 'To', 'Time']
dft = pd.read_csv("sx-mathoverflow.txt", sep = ' ', names = name_row)
dft.head(20)
```

```
Out[89]:
```

	From	To	Time
0	1	4	1254192988
1	3	4	1254194656
2	1	2	1254202612
3	25	1	1254232804
4	14	16	1254263166
5	1	16	1254271943
6	22	16	1254272423
7	1	2	1254273043
8	3	16	1254273303
9	2	1	1254274810
10	27	1	1254275440
11	13	16	1254291078
12	1	28	1254304365
13	21	1	1254384983
14	1	7	1254392595
15	32	1	1254395022
16	7	27	1254396925
17	32	1	1254397771
18	32	27	1254401224
19	1450	22	1254436186

```
In [90]: dft.From.nunique()
```

```
Out[90]: 19774
```

Graph

```
In [91]: wt = nx.from_pandas_edgelist(dft, source = 'From', target = 'To', edge_attr = 'Time', c
```

```
In [92]: nx.info(wt)
```

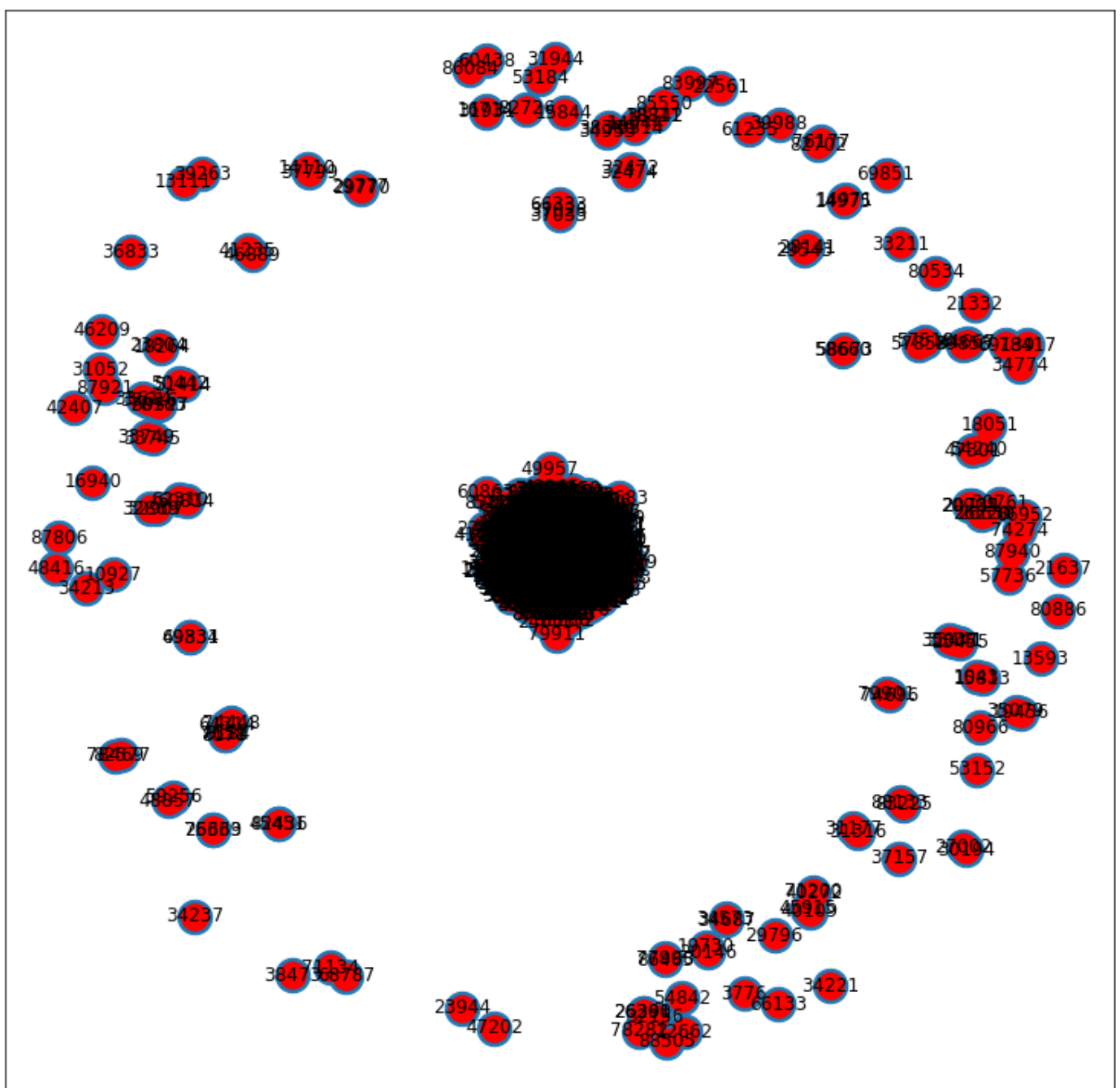
```
Out[92]: 'Name: \nType: Graph\nNumber of nodes: 24818\nNumber of edges: 199973\nAverage degree: 16.1152'
```

```
In [94]: time = nx.get_edge_attributes(wt, 'TIME')
```



```
In [95]: DIG2 = nx.Graph(wt)
        posi = nx.spring_layout(DIG2)
```

```
In [96]: plt.figure(figsize = (13, 13))
        nx.draw_networkx_edge_labels(wt, pos = posi, edge_labels = time, font_size = 12)
        nx.draw_networkx(wt, posi, with_labels = True, node_color = 'r', edge_color = 'g', arro
        plt.show()
```

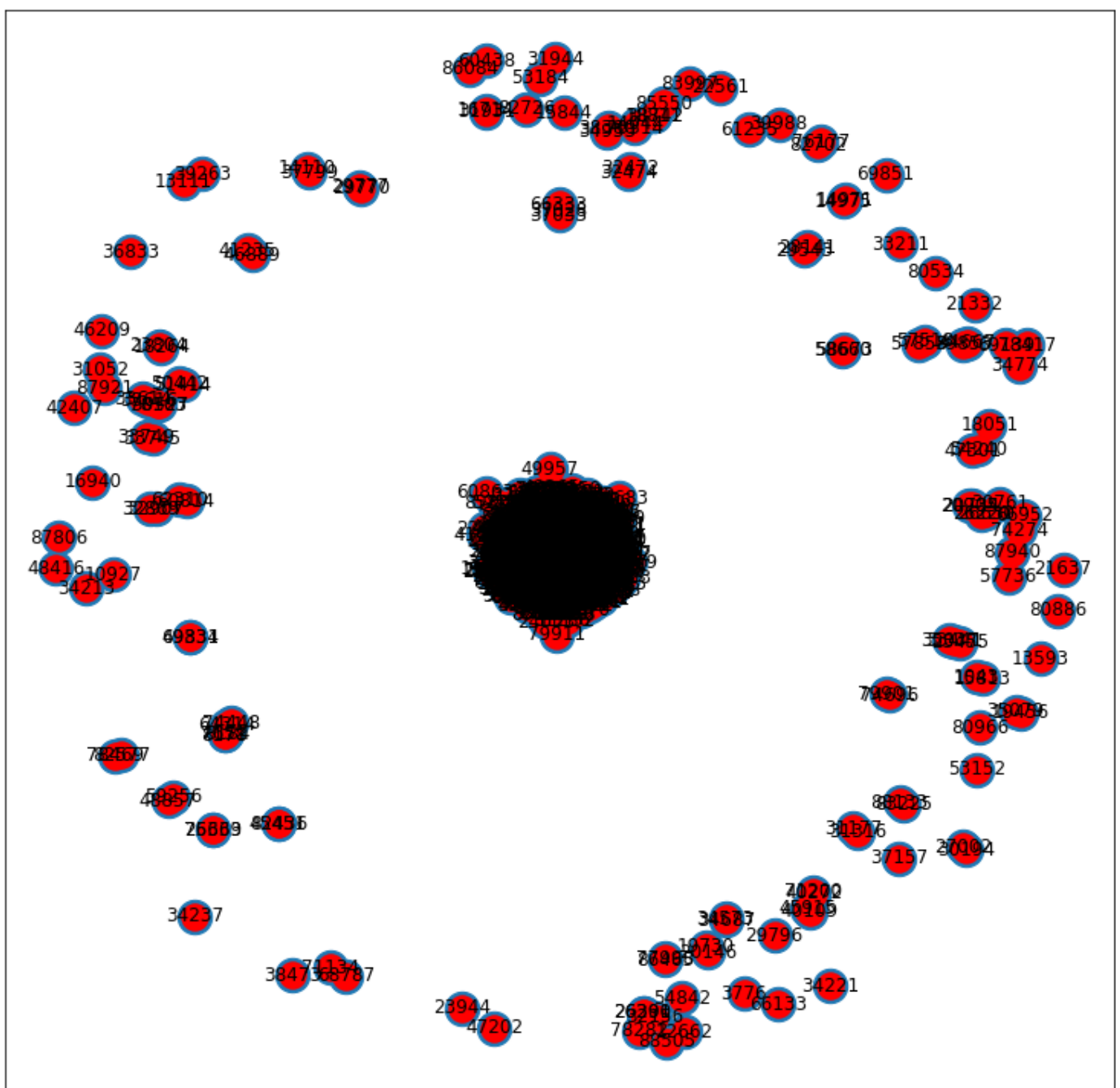



Multi Graph

In [102...

```
wt = nx.from_pandas_edgelist(dft, source = 'From', target = 'To', edge_attr = 'Time', c
plt.figure(figsize = (13, 13))

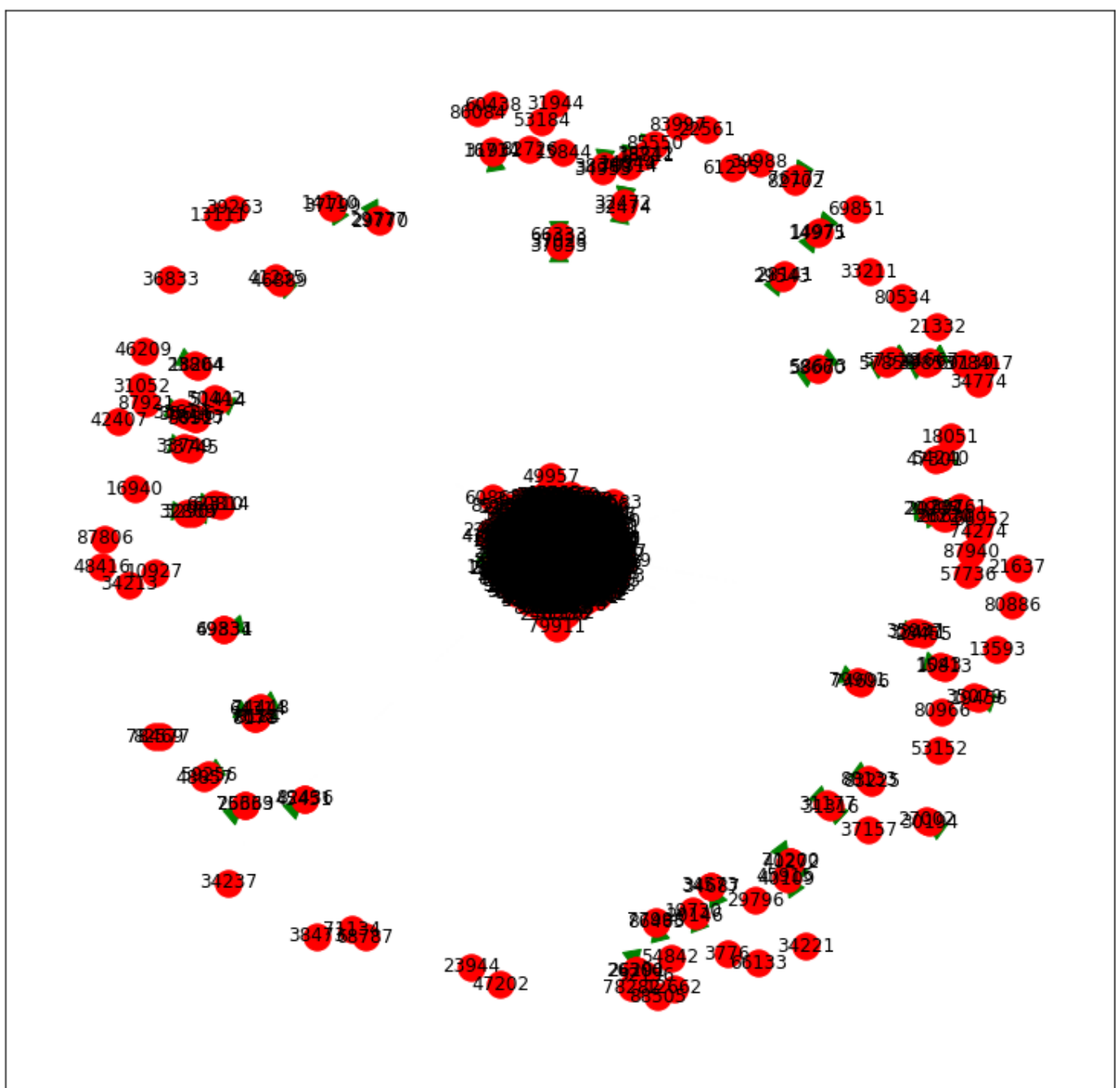
nx.draw_networkx_nodes(wt, posi, node_size = 500)
nx.draw_networkx_edge_labels(wt, pos = posi, edge_labels = time, font_size = 12)
nx.draw_networkx(wt, posi, with_labels = True, node_color = 'r', edge_color = 'g', arro
plt.show()
```



Multi Di Graph

```
In [103... wt = nx.from_pandas_edgelist(dft, source = 'From', target = 'To', edge_attr = 'Time', c
plt.figure(figsize = (13, 13))

nx.draw_networkx_edge_labels(wt, pos = posi, edge_labels = time, font_size = 12)
nx.draw_networkx(wt, posi, with_labels = True, node_color = 'r', edge_color = 'g', arro
plt.show()
```

GIPHE SCREENSHOT:

