Θεωρία Γραφημάτων και Εφαρμογές (Εργαστήριο)
2η Άσκηση

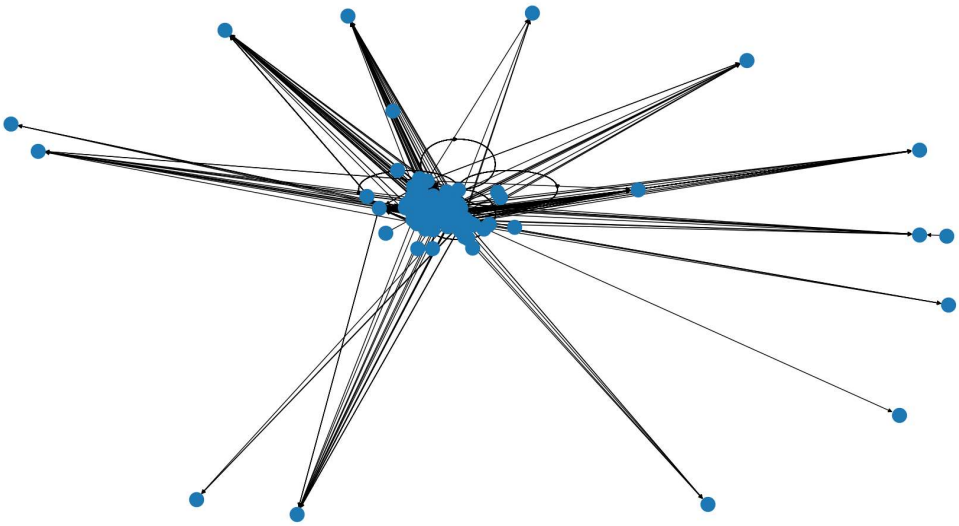
Κώδικας για προετοιμασία της άσκησης:

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


def plot_graph(graph, weight_name=None):
    plt.figure()
    pos = nx.spring_layout(graph)

    if weight_name:
        labels = nx.get_edge_attributes(graph, weight_name)
        nx.draw_networkx_edge_labels(graph, pos, edge_labels=labels)
        nx.draw(graph, pos)
    else:
        nx.draw(graph, pos)
    plt.show()


open_file = open("email_network.txt", "r")
file_text = open_file.read()
open_file.close()
print(file_text)
print()

```

1ο ερώτημα:

```python
graph = nx.read_edgelist("email_network.txt",
                         data=[('time', int)],
                         create_using=nx.MultiDiGraph())
plot_graph(graph)
```

Αποτέλεσμα:



2ο ερώτημα:

```
numEmails = len(graph.edges())
numEmployees = len(graph.nodes())
print("Number of emails: ", numEmails)
print("Number of employees: ", numEmployees)
print()
```

Αποτέλεσμα:

Number of emails: 82927
Number of employees: 167

3ο ερώτημα:

```python
strong = nx.is_strongly_connected(graph)
weak = nx.is_weakly_connected(graph)

print("Is the graph strongly connected? ", strong)
print("Is the graph weakly connected? ", weak)
print()
```

Αποτέλεσμα:
Is the graph strongly connected? False
Is the graph weakly connected? True

4ο ερώτημα:

```python
max_numweak = max(nx.weakly_connected_components(graph), key=len)
print(len(max_numweak))
```

Αποτέλεσμα:
167

5ο ερώτημα:

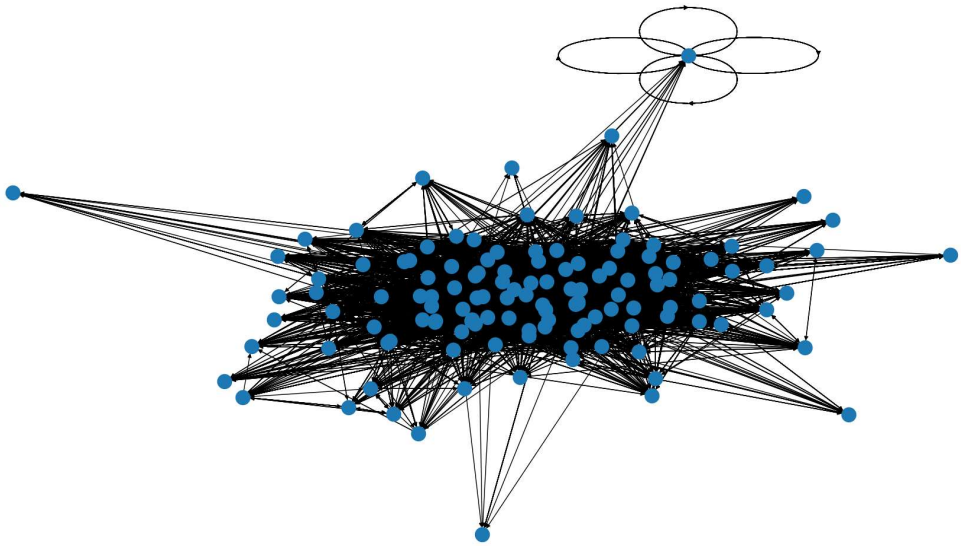
```python
max_numstrong = max(nx.strongly_connected_components(graph), key=len)
print(len(max_numstrong))
```

Αποτέλεσμα:
126

6ο ερώτημα:

```
g = max(nx.strongly_connected_components(graph), key=len)
gsc = graph.subgraph(g)
plot_graph(gsc)
```

Αποτέλεσμα:



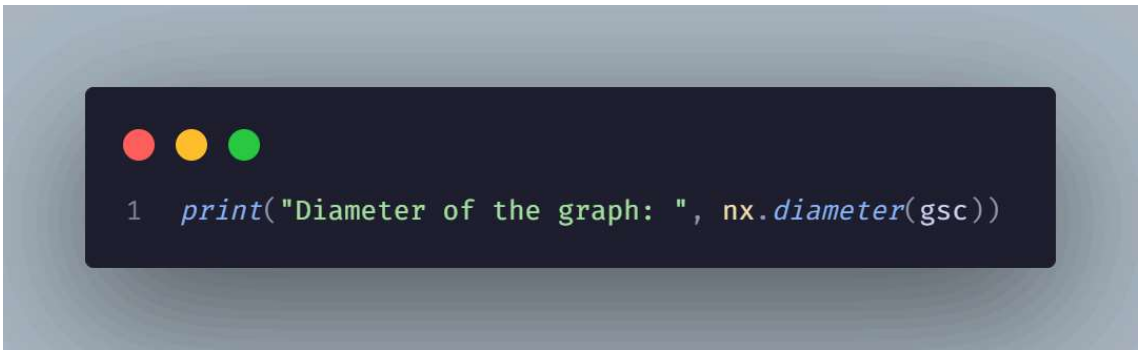
7ο ερώτημα:

```
print("Average distance between nodes: ", nx.average_shortest_path_length(gsc))
```

Αποτέλεσμα:

Average distance between nodes: 1.6461587301587302

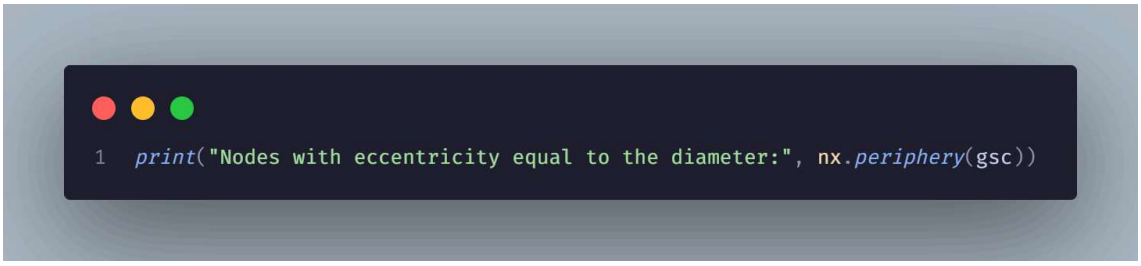
8ο ερώτημα:



```
print("Diameter of the graph: ", nx.diameter(gsc))
```

Αποτέλεσμα:
Diameter of the graph: 3

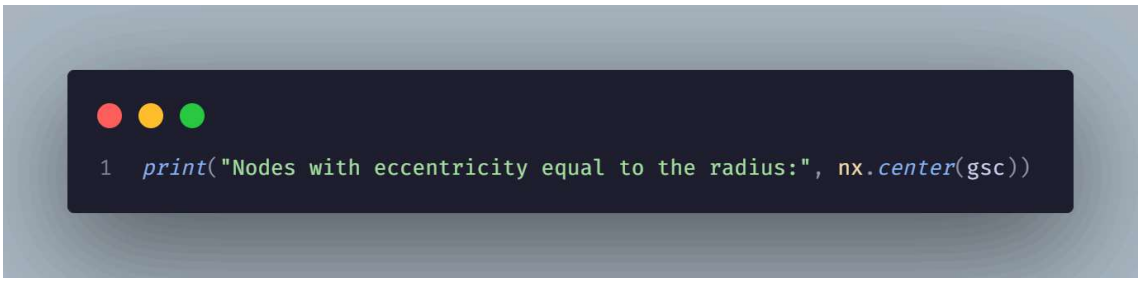
9ο ερώτημα:



```
print("Nodes with eccentricity equal to the diameter:", nx.periphery(gsc))
```

Αποτέλεσμα:
Nodes with eccentricity equal to the diameter: ['97', '129', '134']

10ο ερώτημα:



```
print("Nodes with eccentricity equal to the radius:", nx.center(gsc))
```

Αποτέλεσμα:
Nodes with eccentricity equal to the radius: ['38']

11ο ερώτημα:

```python
peri = nx.periphery(gsc)
diam = nx.diameter(gsc)

numPathsDiam = {}

for node in peri:
    sp = nx.shortest_path(G=gsc, source=node)
    pathsLengthDiam = [
        path for path in sp.values() if (len(path) - 1 == diam)]
    numPathsDiam[node] = len(pathsLengthDiam)

key = list(numPathsDiam.keys())
values = list(numPathsDiam.values())
resultKey = key[values.index(max(values))]

print("Node with the most number of shortest paths equal to the diameter: ",
      resultKey, "Paths: ", numPathsDiam[resultKey])
print()
```

Αποτέλεσμα:
Node with the most number of shortest paths equal to the diameter: 97 Paths: 63

12ο ερώτημα:

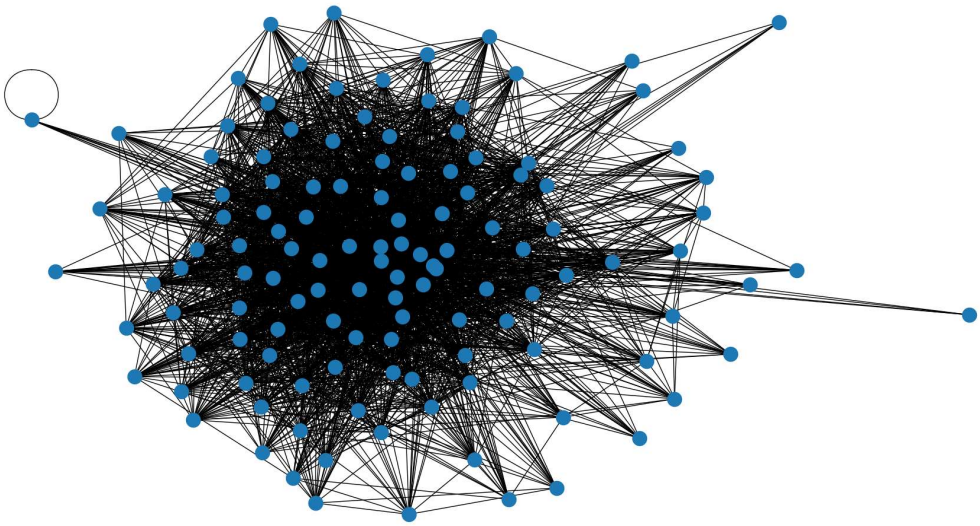
```python
n = resultKey
c = nx.center(gsc)[0]
conn = nx.node_connectivity(gsc, s=c, t=n)
print("Number of nodes required to be removed in order to disconnect the center from the node: ", conn - 1)
```

Αποτέλεσμα:
Number of nodes required to be removed in order to disconnect the center from the node: 5

13ο ερώτημα:

```python
un = gsc.to_undirected()
plot_graph(nx.Graph(un))
```

Αποτέλεσμα:



14ο ερώτημα:

```
g = nx.Graph(un)
print("Transitivity of the graph: ", nx.transitivity(g))
print("Average clustering coefficient of the graph: ", nx.average_clustering(g))
print()
```

Αποτέλεσμα:

Transitivity of the graph: 0.570111160700385
Average clustering coefficient of the graph: 0.6975272437231418