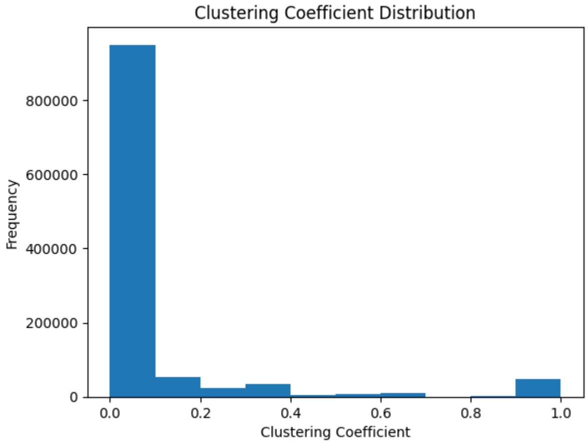
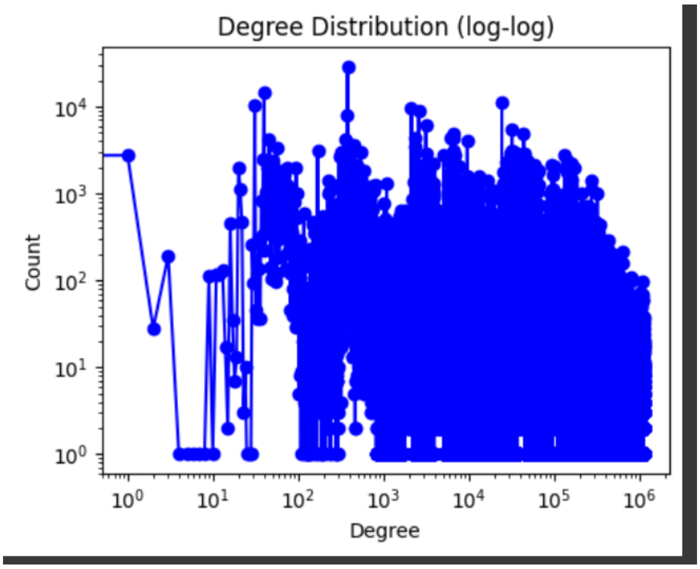
Q7. Social Media Network Analysis

a. Please calculate and plot the clustering coefficient and degree distribution of the network.





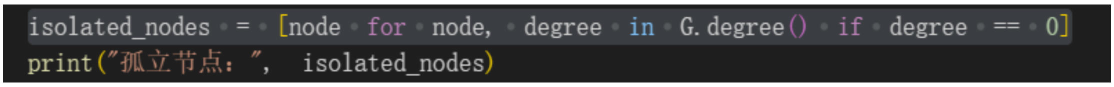
b. Identify the most influential nodes in a network and analyze them. Please use centrality metrics such as degree centrality and then visualize.

最具影响力的节点:1072,363,35661,106,482709

The fig is too large, and it turns black:

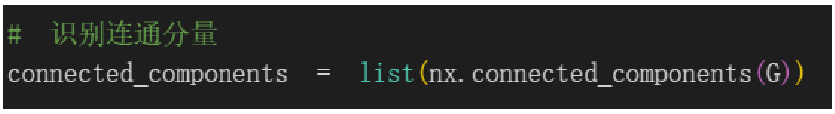
Network Graph

c. Identify Isolated Nodes in the Network
By use



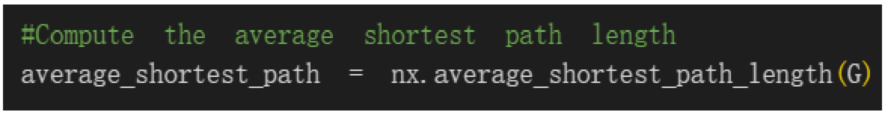
```
isolated_nodes = [node for node, degree in G.degree() if degree == 0]
print("孤立节点：", isolated_nodes)
```

d. Recognize Connected Components in the Network
Buy use



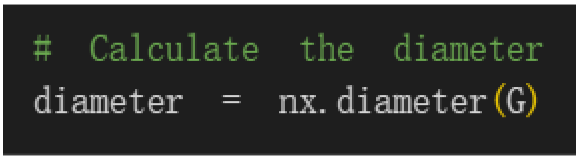
```
# 识别连通分量
connected_components = list(nx.connected_components(G))
```

e. Compute Average Shortest Path Length of the Network.
Buy use



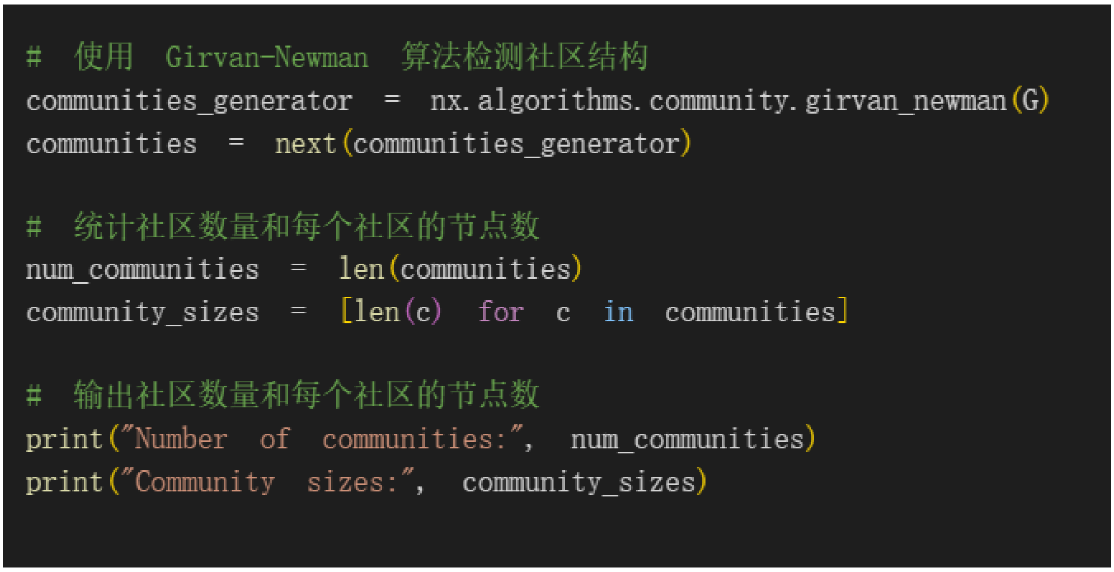
```
#Compute the average shortest path length
average_shortest_path = nx.average_shortest_path_length(G)
```

f. Calculate the Diameter of the Network
Buy use



```
# Calculate the diameter
diameter = nx.diameter(G)
```

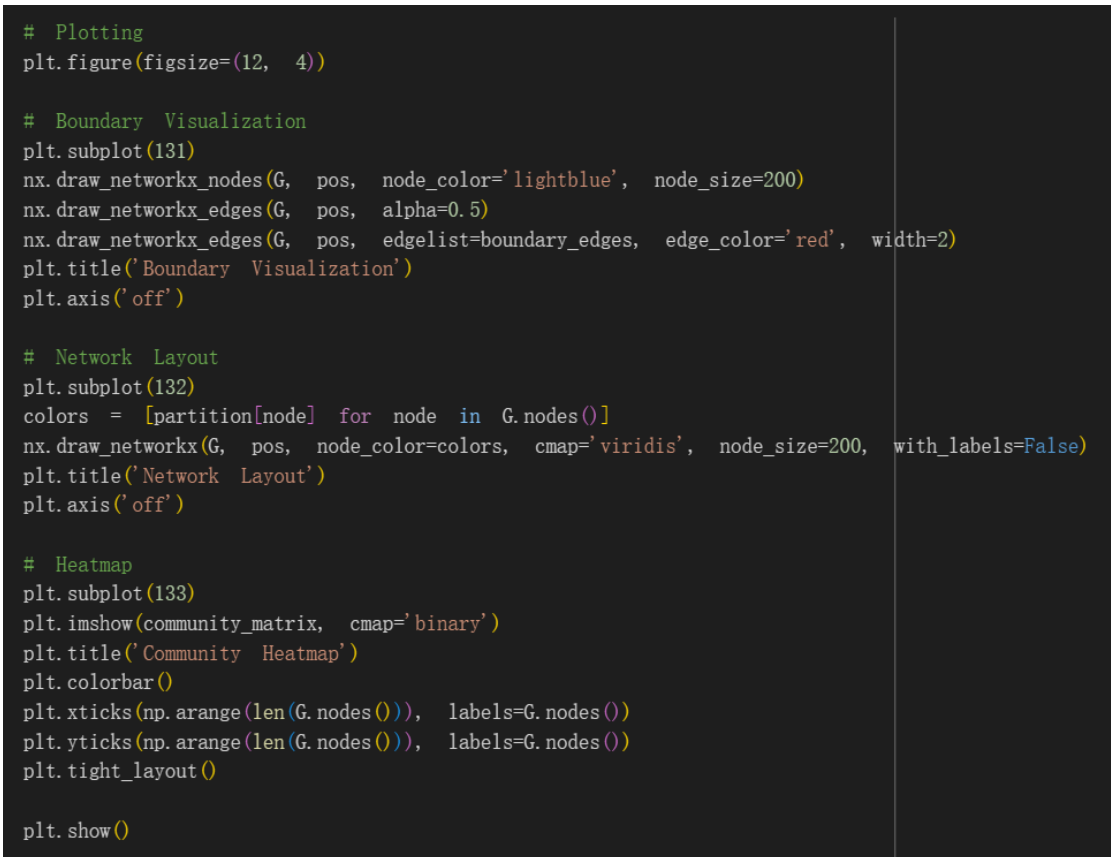
g. Detect Community Structures in the Network.
Buy use



```python
# 使用 Girvan-Newman 算法检测社区结构
communities_generator = nx.algorithms.community.girvan_newman(G)
communities = next(communities_generator)

# 统计社区数量和每个社区的节点数
num_communities = len(communities)
community_sizes = [len(c) for c in communities]

# 输出社区数量和每个社区的节点数
print("Number of communities:", num_communities)
print("Community sizes:", community_sizes)
```



```python
# Plotting
plt.figure(figsize=(12, 4))

# Boundary Visualization
plt.subplot(131)
nx.draw_networkx_nodes(G, pos, node_color='lightblue', node_size=200)
nx.draw_networkx_edges(G, pos, alpha=0.5)
nx.draw_networkx_edges(G, pos, edgelist=boundary_edges, edge_color='red', width=2)
plt.title('Boundary Visualization')
plt.axis('off')

# Network Layout
plt.subplot(132)
colors = [partition[node] for node in G.nodes()]
nx.draw_networkx(G, pos, node_color=colors, cmap='viridis', node_size=200, with_labels=False)
plt.title('Network Layout')
plt.axis('off')

# Heatmap
plt.subplot(133)
plt.imshow(community_matrix, cmap='binary')
plt.title('Community Heatmap')
plt.colorbar()
plt.xticks(np.arange(len(G.nodes())), labels=G.nodes())
plt.yticks(np.arange(len(G.nodes())), labels=G.nodes())
plt.tight_layout()

plt.show()
```

The data is too many to visalize. I feel very sorry about it. It need to many time.