**社交网络分析课程实验报告1**

姓名： 学号：

请修改文件名称后提交！

网络数据已发到QQ群，数据中每个数字表示网络中一个节点，每行表示以两节点为端点的一条边。注意：(1) 网络中部分节点存在自环；(2) 网络中节点标号从0开始，未必连续，因此最大数字和节点数量关系不确定；(3) 数据首行”id\_1,id\_2”为提示信息，建议删除，或在程序中跳过。

问题1：统计网络中的节点数量*n*和边的数量*m*（边包括自环），并计算网络密度*d*（边数包括自环，保留两位小数，公式如下）；



问题2：计算网络的平均聚集系数*c*（保留两位小数，公式参见教材，节点度不包含自环）。

需要回答以下问题：

1. 程序结果

节点数量*n* = 22470

边的数量*m* = 171002

网络密度*d* = 0.00068

网络平均聚集系数*c* = 0.36

2. 编程语言（如Python, C, C++）

Python

3. 实验环境（如PyCharm XXX, CodeBlocks XXX, Visual Studio XXX）

PyCharm 2023.3.3

4. 代码（缩进清晰，需提供必要注释）

import pandas as pd

import networkx as nx

# 读取CSV文件，跳过首行提示信息

data = pd.read\_csv('musae\_facebook\_edges.csv', skiprows=0)

data.columns = ['id\_1', 'id\_2']

# 创建无向图

G = nx.from\_pandas\_edgelist(data, 'id\_1', 'id\_2', create\_using=nx.Graph())

# 添加自环，如果有的话

self\_loops = data[data['id\_1'] == data['id\_2']]

G.add\_edges\_from(self\_loops.values)

# 计算节点数量n

n = G.number\_of\_nodes()

# 计算边的数量m（包括自环）

m = G.number\_of\_edges()

# 计算网络密度d

d = (2 \* m) / (n \* (n - 1))

# 计算网络的平均聚集系数

average\_clustering = nx.average\_clustering(G)

# 输出结果

print(f'节点数量n = {n}')

print(f'边的数量m = {m}')

print(f'网络密度d = {d:.5f}')

print(f'网络平均聚集系数c = {average\_clustering:.2f}')