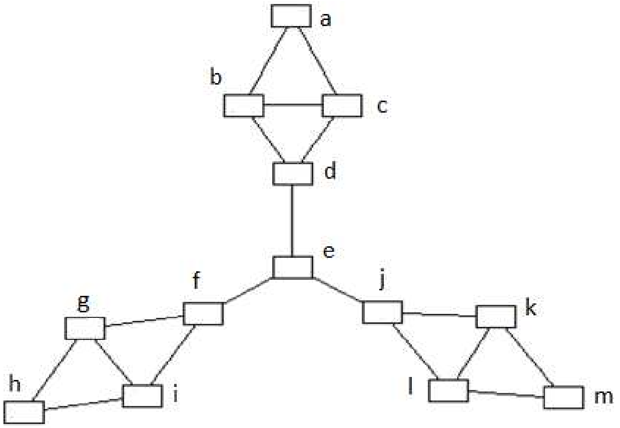
Homework 1 (10 points, Due Feb 15)

[Please submit your homework through Blackboard]

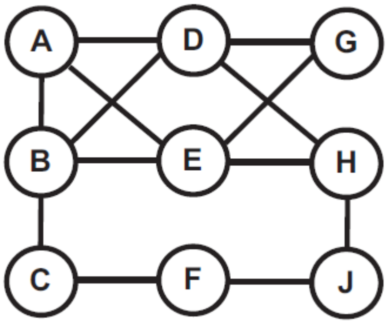
Question 1 (2.0 points: 0.25/each): Please use your own language to briefly explain the following concepts:

Social networks:
Undirected graph:
Adjacency matrix:
Network Diameter:
Centrality score of a node:
Random Walk:
Random Graph:
Power-Law Distribution:

Question 2 (1.5 points): In the following network, please calculate the Betweenness Centrality scores [0.5 pt], Closeness Centrality score [0.5 pt], and Eigen Vector based centrality scores [0.5 pt] for every nodes in the network (please show your solutions)

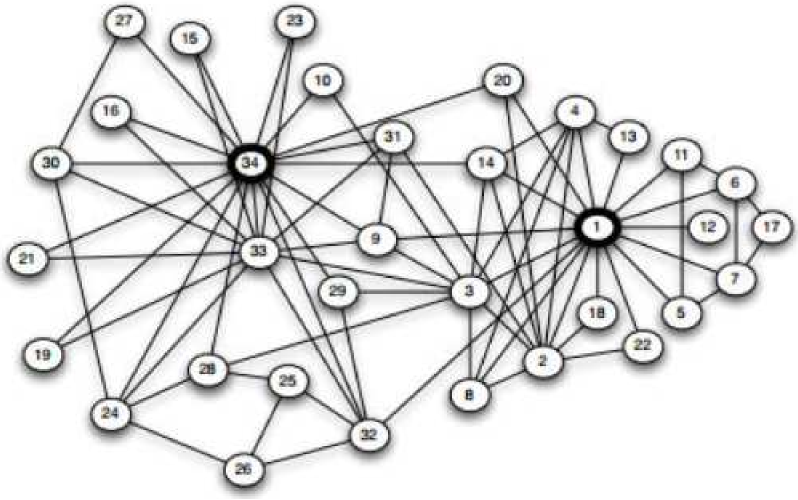


Question 3 (2.5 pts): In the following network, please explain how to use adjacency matrix and the power of adjacency matrix to find diameter of the network (show your solution 1 pt). Please draw degree distribution of the network [0.25 pt], calculate clustering coefficient for very nodes in the network [0.25 pt]. Please also calculate the edge density [0.25 pt] and the clustering coefficient of the whole network [0.25]. Please explain why clustering coefficient is smaller than the edge density [0.25 pt]. Please find the node with the highest betweenness score (please show your solution [0.25 pt])



Question 4 (2.0 pts) The following network shows a small benchmark “Zachary's karate club” social network which contains “friendships between 34 members of a karate club at a US university in the 1970s”.

1. Please calculate the edge density and the clustering coefficient of the whole network, and analyze why clustering coefficient is larger (or smaller) than the edge density [0.5 pt].
2. Please calculate the average distance between any two pairs of nodes [0.5 pt], and report the Diameter of the network [0.5 pt].
3. Please find the node(s) with the highest closeness centrality score [0.5 pt]



Zachary’s karate club network

Question 5 (2.0 points) The following URL points to a "coauthorship network" of scientists working on network theory and experiment.

http://networkdata.ics.uci.edu/data.php?id=11

A brief description of the network is given in the "netscience.txt". In "netscience.paj" file (these are text files), you can find nodes and edges between nodes. The names of the scientists (which correspond to the nodes of the networks) are given in "netscience.gml". Please download the dataset and use any program tools you are familiar with to build a network and finish the following tasks:

4. Please report and draw the degree distributions of the whole network [0.25 pt]. Convert the figure to log-log space and validate whether it complies with the power-law distributions [0.25 pt].
5. Please report cumulative degree distribution of the network [0.25 pt], and convert it to log-log space and validate whether it complies with the power-law distributions [0.25 pt].
6. Please report Rank-Degree distribution of the network [0.25 pt], and convert it to log-log space and validate whether it complies with the power-law distributions [0.25 pt].
7. Please report clustering coefficient and diameter of the network [0.5 pt].

Tips for programming:

In order to calculate the average distance between a pair of nodes, and calculate the diameter of the network, you will need to implement algorithms which calculate shortest path between any two nodes. Example of algorithms include

Dijkstra's algorithm (http://en.wikipedia.org/wiki/Dijkstra's_algorithm)

Breath First Search algorithm (http://en.wikipedia.org/wiki/Breadth-first_search)

Alternatively, you may consider following programming tools/packages which are specifically designed for network and graph data. These packages have algorithms for finding shortest path and network diameter.

Gephi: The open Graph Viz Platform

https://gephi.org/

(Please note Gephi also has API functions to support user programming. You can check API functions for the following URL:

https://gephi.org/docs/api/

Java Platform: JUNG (Java Universal Network/Graph Framework)

http://jung.sourceforge.net/

http://www.datalab.uci.edu/papers/JUNG_tech_report.html

Python: NetworkX (High-productivity software for complex networks)

http://networkx.github.com/

.Net: NodeXL (Open source template for Microsoft tools)

http://nodexl.codeplex.com/