Degree Distribution:

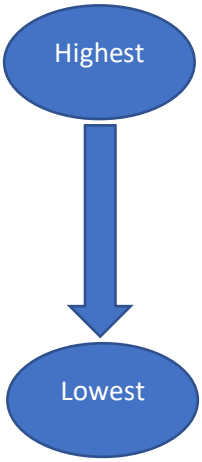
1. Generate a few random graphs. You can do this using networkx's random graph generators. Do the random graphs you tested appear to be scale free? (Include degree distribution with your answer).
   For Random graphs:
   gnm1 : Scale Free
   gnm2 : Not Scale Free
   gnp1 : Not Scale Free
   gnp2 : Not Scale Free

2. Do the Stanford graphs provided to you appear to be scale free?
   For Stanford graphs:
   amazon.graph.large : Not Scale Free
   amazon.graph.small : Scale Free
   dblp.graph.large : Not Scale Free
   dblp.graph.small : Not Scale Free
   youtube.graph.large : Not Scale Free
   youtube.graph.small :Not Scale Free

Centrality:

1. Rank the nodes from highest to lowest closeness centrality.

id	closeness
F	0.07142857142857142
C	0.07142857142857142
H	0.06666666666666667
D	0.06666666666666667
B	0.058823529411764705
E	0.058823529411764705
G	0.05555555555555555
A	0.05555555555555555
I	0.047619047619047616
J	0.034482758620689655



2. Suppose we had some centralized data that would sit on one machine but would be shared with all computers on the network. Which two machines would be the best candidates to hold this data based on other machines having few hops to access this data?
   F and C would be the two best candidates to hold this data based on other machines having few hops to access this data

Articulation Points:

1. In this example, which members should have been targeted to best disrupt communication in the organization?

   It would have been best to target the following members to disrupt communication in the organization:
   - Mohamed Atta
   - Usman Bandukra
   - Mamoun Darkazanli
   - Essid Sami Ben Khemais
   - Djamal Beghal
   - Nawaf Alhazmi
   - Raed Hijazi

Note:

- The code to check if the graph is scale free or not is in the file "power_law.py"
- Example to run this file:

  python power_law.py filename.csv

- Here filename.csv is the output file create by "degree.py" file