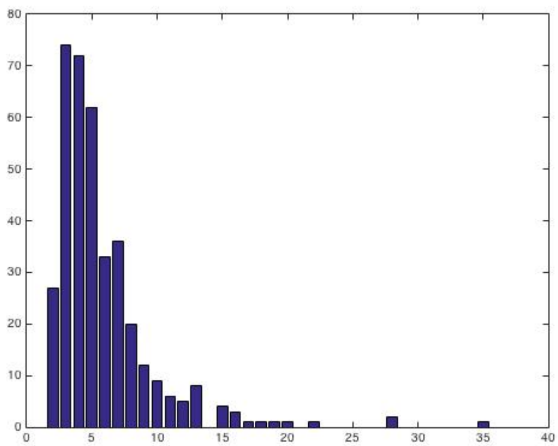
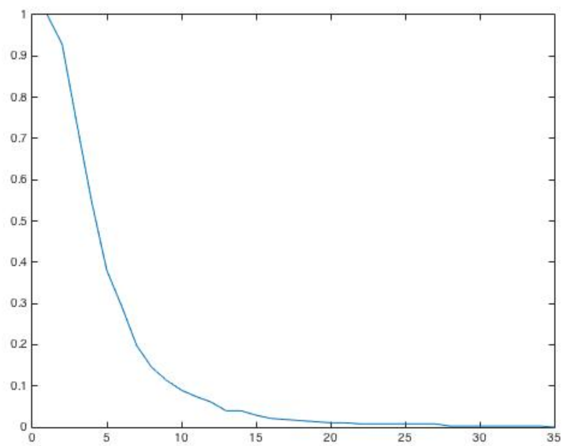
Problem2:

Code for this problem is problem2.py

The outcome of the problem.py is documented in degree.txt as well as shown in the terminal screen. For the histogram and the ccdf of the node degrees, I use the documented degree.txt file as an input and plot it in Matlab.

1. The histogram and ccdf of the node degrees:





2. average clustering coefficient: 0.741230614293
   overall clustering coefficient: 0.430575035063

3. maximal diameter: 17
   average diameter: 6.04186734794

Algorithm:

1. when computing the degree distribution, I just go through each node to calculate its edges. The time complexity of this algorithm is O(n).
2. when computing the clustering coefficient of the graph. I go through each node to find how many triangles it has. The time complexity of this algorithm is:

$$\sum_{k=1}^{n}\sum_{j=0}^{d_k-1} j = \sum_{k=1}^{n}\frac{d_k(d_k-1)}{2} \approx \sum_{k=1}^{n}\frac{d_k^2}{2}$$
$$\leq \mathrm{d_{max}}\sum_{k=1}^{n}\frac{d_k}{2} = d_{max}\times m$$

   where $\mathrm{d_{max}}$ is the maximal degree, and m is the amount of edges.
3. When computing the diameter, I use the Floid Algorithm. First I construct a Matrix with

shows the connectivity among nodes. If i node connects with j node, then A_{ij} is 1, otherwise, A_{ij} is n(a comparative large number). Then I iterately compute the shortest distance of each pair of two nodes and continuously update the shortest distance to the Matrix. The time complexity of this algorithm is O(n^3).