Large network structure and random graph models

You are given a **selection of real networks** of different sizes and origins.

- Zachary's karate club network (karate.net, 34 nodes)
- Davis's southern women network (women.net, 32 nodes)
- Lusseau's bottlenose dolphins network (dolphins.net, 62 nodes)
- Ingredients network by common compounds (ingredients.net, 1,525 nodes)
- Map of Darknet from Tor network (darknet.net, 7,178 nodes)
- Human protein-protein interaction network (ppi.net, 19,634 nodes)
- Internet map of autonomous systems (internet.net, 75,885 nodes)
- Amazon product copurchase network (amazon.net, 262,111 nodes)
- Paper citation network of APS (aps.net, 438,943 nodes)
- Small part of Google web graph (google.net, 875,713 nodes)
- Road/highway network of Texas (texas.net, 1,379,917 nodes)

All networks are available in Pajek format.

I. Toy network construction and Pajek format

1. *(starter)* Using your library, **construct small toy network** with a few nodes and edges. Print out its name, and the number of nodes and edges.

2. Using the methods provided by your library, **read all real networks** and print out their size. What is the

size of the largest network you can read in say half a minute?

II. Network statistics, connectivity, distances and clustering

1. Compute **basic statistics of networks**. These are the number of nodes n, the number of isolated nodes n_0, the number of edges m, the number of self-edges or loops m_0, the average node degree $\langle k \rangle = 2m/n$ and the maximum degree k_{max}, and the undirected density $\rho = \langle k \rangle/(n-1)$. Are the results expected?

   Computational complexity is $\leq$ linear $\mathcal{O}(m)$ and applicable to any network that fits in your memory.

2. Using depth-first search methods provided by your library, compute **connected components of networks**. Print out the fraction of nodes in the largest connected component S and the number of all connected components s. Are the results expected?

   Computational complexity is linear $\mathcal{O}(m)$ and applicable to any network that fits in your memory.

3. Using breadth-first search methods provided in the script, approximate **distances between the nodes of networks**. Print out the average distance between the nodes $\langle d \rangle$ and the maximum distance or diameter d_{max}. Are the results expected?

   Computational complexity is quadratic $\mathcal{O}(nm)$ and applicable only to medium sized networks.

4. Using triad counting methods provided by your library, compute **clustering coefficient of networks**. Print out the average clustering coefficient $\langle C \rangle$. Are the results expected?

   Computational complexity is superlinear $\mathcal{O}(m\langle k \rangle)$ and applicable to all but the largest networks.

5. *(tentative)* Using plotting functionality provided by your library, compute **degree distribution of networks**. Plot degree distribution p_k on a doubly logarithmic plot. Are the results expected?

   Computational complexity is linear $\mathcal{O}(n)$ and applicable to any network that fits in your memory.

6. What is the **size of the largest network** you can analyze in say half a minute?

III. Random graphs and standard network models

1. Using the methods provided by your library, construct **Erdös-Rényi random graphs** $G(n, m)$ with the same number of nodes n and edges m as the networks above. Print out their basic statistics. Are the results expected?

2. *(tentative)* Using the methods provided by your library, construct **Barabási-Albert scale-free networks** $G(n, \langle k \rangle/2)$ with the same number of nodes n and the average degree $\langle k \rangle$ as the networks above. Print out their basic statistics. Are the results expected?