Complex Systems 535/Physics 508: Homework 4

1. (a) You are given a road map and told the average driving time along each road segment, then you are asked to find the route from A to B with the shortest average driving time. What algorithm would you use to do this, and what would be the time complexity of the calculation?

   (b) Given an undirected network, what algorithm would you use to determine whether there are at least two node-independent paths between a given pair of nodes?

   (c) What algorithm would you use to find all the components in an undirected network, and what would be the time complexity of the operation?

2. For a directed network in which in- and out-degrees are uncorrelated, show that it takes time $O(m^2/n)$ to calculate the reciprocity of the network. Why is the restriction to uncorrelated degrees necessary? What could happen if they were correlated?

3. Suggest one potential source of error in measurements of the structure of each of the following networks:

   (a) A scientific coauthorship network assembled from a database of papers.

   (b) A web network of web pages at a single university, assembled by using an automated web crawler.

   (c) A metabolic network.

   (d) A social network of who is friends with whom at a large company, assembled using questionnaires.

   (e) A network representation of an electrical power grid.

4. Suppose we draw n independent random reals x in the range $0 \leq x < \infty$ from the (properly normalized) exponential probability density $P(x) = \mu e^{-\mu x}$.

   (a) Write down the likelihood (i.e., the probability density) that we draw a particular set of n values x_i (where $i = 1 \ldots n$) for a given value of the exponential parameter μ.

   (b) Maximize your expression to find a formula for the maximum-likelihood estimate of μ given a set of observed values x_i.

5. **Extra credit programming challenge:** *This question is optional, for extra credit. You can get 100% on this week's homework without doing it.*

   The file `http://www.umich.edu/~mejn/courses/2017/cscs535/internet.txt` contains a snapshot of the structure of the Internet at the autonomous system level in adjacency list format. The $n = 22\,963$ vertices in the network are numbered in order from zero (note: not from 1) up to 22962, with one line of the file for each one. A single line of the file looks, for example, like this:

   ```
   112 3 12 22 24
   ```

   This means that vertex 112 has degree 3, and its three neighbors are the vertices numbered 12, 22, and 24.

   Write a program in the programming language of your choice to do the following:

(a) Read the data from the file into a data structure of your choice, to represent the network, in adjacency list format, in the memory of the computer. For example, in C you might use a dynamically allocated 2D integer array with a line for each vertex, and another 1D array for the degrees. In Python you might use an array of n lists or sets to store the neighbors of each vertex, and an array of integers for the degrees.

(b) Write code to calculate, using breadth-first search, the shortest distance from a given single vertex to every other, then average those distances to calculate the closeness centrality of the given vertex.

Use your program to calculate the closeness centrality of the vertex numbered 0 in the network.

For full extra credit, turn in a complete printout of your program, and a printout of it in action, showing the answer it finds.