The Structure of Information Networks	**Homework 1**
CS 6850 Spring 2011	**Due Friday, February 11, 2011**

The goal of this problem set is to provide practice implementing some basic network analysis techniques on a moderate-sized network dataset — specifically, a coauthorship network constructed from a bibliography of computer science papers maintained by Joel Seiferas at the University of Rochester. The bibliography can be downloaded from

```
ftp://ftp.cs.rochester.edu/pub/u/joel/papers.lst
```

Building the coauthorship network. Here are some instructions on how to create the coauthorship network from the raw bibliography. The short explanation is: each line represents a paper, and we want to build the undirected graph whose nodes are the people named in the bibliography, and whose edges join those pairs of people who've coauthored a paper in the bibliography.

The more detailed instructions now follow. Each line in the bibliography describes a distinct paper, and has the following format:

```
year [number] conference/journal author & author & ...  & author, title
```

Here, `conference/journal` is an acronym encoding the conference or journal where the paper appeared, `year` is the year of the paper, and `number` is the volume number of the journal or conference. We write `number` in brackets above because it is present in some lines and (when it is not known or not applicable) absent in others. Authors are given by last name only, and separated by the & symbol. The list of authors ends with a comma, and the the remainder of the line is the title. Thus, a sample line from the file is

```
2005 37 STOC Naor & Schwartz, Balanced Metric Labeling
```

encoding the paper "Balanced Metric Labeling" by Naor and Schwartz at the 37th STOC conference in 2005. Finally (as within list of records of this length), it is possible that a few of the lines in the file are misformatted.

From this bibliography, you should construct a coauthorship network as follows.

- There should be one node for each person. (Note that even if a person is an author on 50 of the papers listed in the bibliography file, there should still just be one node corresponding to him or her, not 50.)
- There should be an undirected edge between nodes A and B if and only if they are coauthors on a paper in the bibliography. (If they are coauthors on multiple papers, there should still just be a single edge joining them.)

For example, if the file consisted of just the three lines

```
2005 37 STOC Naor & Schwartz, Balanced Metric Labeling
2005 37 STOC Alon & Shapira, Every Monotone Graph Property is Testable
1996 45 IEEETC Azar & Naor & Rom, Routing Strategies for Fast Networks
```

then the graph should have node set

$$\{Alon, Azar, Naor, Rom, Schwartz, Shapira\}$$

and edge set

$$\{(Alon, Shapira), (Azar, Naor), (Azar, Rom), (Naor, Rom), (Naor, Schwartz)\}.$$

Caveats. Before we move on to the problems themselves, here are two points worth mentioning about the network we're studying here.

(1) As we'll see at various points in the course, coauthorship networks are a popular kind of "model system" for large-scale network analysis. This is not so much because there's widespread fascination with the coauthoring habits of scientists (though it's an interesting topic that some people study as their research area), but because coauthorship networks are a kind of social network, encoding a particular type of collaboration among people, for which extremely rich and detailed data is available. As a result, it is a chance to try out network analysis techniques at very high resolution, in a setting that possesses many of the properties exhibited by much "messier" and harder-to-measure social networks as well.

(2) Any time one tries to build a network from a file containing a list of names, there's the concern that different people can have the same name, and hence these different people are being "merged" into a single node. This is definitely something to worry about when one tries to draw inferences about social structure from the resulting network. However, in our case, we are using this dataset simply to build an interesting graph on which to practice various analysis techniques, so for our limited purposes there's no problem: if two authors have the same last name, then for us they *are* the same person.

(2′) In fact, because of the issue in (2), there are papers where someone appears to coauthor with themselves. We will omit from the network those edges that link some node to itself.

What to hand in

You should upload the following files to CMS; please read this section carefully, since the format is important. In particular, for the first file, there is a specific line format we need, since we will be using scripts as part of the grading.

The files to hand in:

(1) An ascii .txt file named "`hw1solution.txt`". This should have results for the questions below, with each line on which you are reporting part of the answer beginning with a "`@`". The form for these lines will be described in the questions below.

(2) Three files named "`plot1`", "`plot2`", and "`plot 3`" containing the plots associated with Questions 1, 2, and 3 respectively. These can be in any standard image format (e.g. `plot1.png`, or `plot1.jpg`, and so forth.).

(3) The source code you used to compute the answers. By default, we won't be grading the quality of the code itself, but it will be useful to have it in case we run into any confusion.

It is fine to use packages or software specifically designed for handling graphs, in which case you should just include what you wrote. If you answer the questions by some means where the notion of "source code" doesn't exactly apply, such as an interactive session with a software package, then submit whatever analogue of source code we'd need to see how you answered the questions — for example, a script you wrote as part of some larger existing package, or a transcript of an interactive session in which you did it. (This file should be named "`code`" and can be in any format; if you need to bundle together multiple files, for example, it can be a zipped or gzipped folder or tar file.)

(4) A brief description of how to apply your code (or code analogue, or transcript) to the data, together with any decisions you made about how to handle the data that would be useful for us to know about. (This file should be named "`explanation`" and can be in any format.)

Again, for most of the solutions, we'll simply be evaluating (1) and (2), and only consulting (3) or (4) as background if necessary.

The Problems

(1) Recall that the *degree* of a node is the number of edges it's incident to. We start by considering how the degrees of the nodes are distributed.

Thus, for a number j, let n_j denote the number of nodes with degree exactly j. Let d^* be the maximum degree of any node in the network. (This is the maximum total number of co-authors that any one author has — the maximum j for which $n_j > 0$.)

(a) For each j from 0 to d^*, output the number n_j. Each of these should correspond to a line in the file `hw1solution.txt` with the following four fields

`@ 1` j n_j

(The second field here simply denotes that you're answering the first question.) So for example, in the file above consisting only of the three lines

```
2005 37 STOC Naor & Schwartz, Balanced Metric Labeling
2005 37 STOC Alon & Shapira, Every Monotone Graph Property is Testable
1996 45 IEEETC Azar & Naor & Rom, Routing Strategies for Fast Networks
```

the correct output would be

```
@ 1 0 0
@ 1 1 3
@ 1 2 2
@ 1 3 1
```

(b) Produce a scatterplot in the plane of the ordered pairs $(\log j, \log n_j)$ for those j such that both $j > 0$ and $n_j > 0$. Hand this in as the file `plot1`. Later in the course, we'll see some proposed explanations for why such scatterplots can often be approximated fairly well by a straight line.

(2) Now we consider the sizes of the connected components in the network.

(a) Let n^* be the number of nodes in the largest connected component, and let n be the number of nodes in graph overall. Report these two quantities and their ratio, as a line in the file `hw1solution.txt` of the form

`@ 2` n^* n n^*/n

For example, on our sample graph above, you would report

```
@ 2 4 6 .667
```

Looking at the ratio of these two quantities is a good way to assess whether we should think of the network as having a "giant" component, or whether it consists entirely of small components.

(b) Let k_j denote the number of connected components of size j, and let c^* denote the size of the second-largest component. For each j from 1 to c^*, output the number k_j. Each of these should correspond to a line in the file `hw1solution.txt` with the following four fields

`@ 2` j k_j

For example, on our sample graph above, you would report

```
@ 2 1 0
@ 2 2 1
```

(c) Produce a scatterplot in the plane of the ordered pairs $(\log j, \log k_j)$ for those j such that both $1 \leq j \leq c^*$ and also $k_j > 0$. Hand this in as the file `plot2`. The extent to which logarithmic plots of component sizes should look like straight lines is less heavily studied, but there is evidence for this as well.

(3) We next consider node-to-node distances in the largest component.

(a) We start by fixing the author name `Hartmanis` (i.e. Juris Hartmanis, one of Cornell's two Turing Award winners) as our "root node." For each j, let r_j denote the number of nodes at distance exactly j from `Hartmanis`. (So $r_0 = 1$, and r_1 is equal to the degree of `Hartmanis`.) Let s^* denote the largest j for which $r_j > 0$ — this is the farthest anyone in the bibliography is from `Hartmanis`, yet still connected to him by a path.

For each j from 1 to s^*, output the number r_j. Each of these should correspond to a line in the file `hw1solution.txt` with the following four fields

`@ 3` j r_j

For example, on our sample graph above, if the starting node were `Schwartz` (rather than `Hartmanis`), you would report

```
@ 3 1 1
@ 3 2 2
```

(b) Produce a histogram that plots r_j as a function of j, for j from 1 to s^*. Hand this in as the file `plot3`.