525 Computing Project 1, Fall 2011*
It's a small world, after all

In the 1960s, Stanley Milgram conducted an experiment to determine the diameter of the social network of the united states. Beginning with a few people in Kansas and Nebraska, participants were asked to mail a package to someone in Boston, but they could only do so by mailing it to people they new. Surprisingly, most of the packages only had to be mailed three times to reach their end point. Later experiments found that to reach a foreign country, the number of hops was usually at most six. This gave rise to the expression "six degrees of separation."

In this project, we will study shortest path problems and see how loosely connected graphs can have very small diameter.

Recall that a *graph* $\mathcal{G}$ is just a collection of nodes $\mathcal{N}$ and links between them which we call *edges*, $\mathcal{E}$. There are two matrices of interest. $\mathcal{I}$ is the node-incidence matrix and $\mathcal{A}$ is the adjacency matrix. K will denote the average degree of a vertex and N will denote the number of nodes. Use the values $N = 1000$ and $K = 10$ throughout.

1. Write a general program called `shortest_path` which takes three arguments: an adjacency matrix of an undirected graph $\mathcal{A}$, and two nodes i_1 and i_2 in the range $1, \ldots, N$ and returns the shortest path from node i_1 to node i_2. Use a linear programming solver of your choice, but do this with linear programming, not using a combinatorial algorithm. The syntax should be

   ```
   path = shortest_path(A,i,j);
   ```

   Note that in this case, you should convert your adjacency matrix into an edge incidence matrix in your code.

*Due in class, November 21, 2011

2. Generate a graph $\mathcal{G}_0 = (\mathcal{N}, \mathcal{E})$ by the following rule: node i is connected to node j if $|i-j| < \frac{K}{2}$ or $|i-j| > N - \frac{K}{2}$. What is the degree of each node in this graph?

   Plot the resulting graph as follows: place node k at the point

   $$\left(\cos\left(\frac{2\pi k}{N+1}\right), \sin\left(\frac{2\pi k}{N+1}\right)\right)$$

   draw an edge between adjacent nodes using the Matlab function `line`.

3. Compute the length of the shortest path from node 1 to node $N/2$ and from node $N/4$ to node $3N/4$. Plot the resulting path using the same drawing scheme as in Problem 2.

4. Let's now modify the graph to shorten the average path length. For each edge in the graph $\mathcal{G}_0$ rewire it with probability β (here you should assume that i is less than j. Rewiring is done by replacing (i, j) with (i, k) where k is chosen uniformly at random from $\{1, \ldots, N\} \setminus \{i\}$.

   For this step, you need to be able to choose random numbers. The Matlab function `rand` will generate random numbers uniformly distributed in $[0, 1]$. When you need to generate an action with probabilty β, this is the same as performing the action if `rand` $< \beta$. If you need to generate a random integer between 1 and M, you can use the call `ceil`($M *$ `rand`). `ceil` is the ceiling function and returns the smallest integer greater than or equal to its argument.

   For the values of $\beta \in \{0.001, 0.01, 0.05, 0.1, 0.2\}$, compute a modified graph $\mathcal{G}_\beta$. For each β, plot the graph $\mathcal{G}_\beta$ as in Problem 2. Compute the lengths of the shortest paths from node 1 to node $N/2$ and from node $N/4$ to node $3N/4$. Plot these paths as in problem 3. Compute the shortest path 10 times and return the average length shortest path for each value of β.

5. Now generate a completely random graph, $\mathcal{G}_r$, where each node has exactly K neighbors. That is, for node i, pick K points uniformly at random from $\{1, \ldots, N\} \setminus \{i\}$ and assign edges from i to each of these points. For this problem, you may use the function `randperm`(N) which generates a random ordering of the integers between 1 and N. Again, plot this graph. Compute the lengths of the shortest paths from node

1 to node $N/2$ and from node $N/4$ to node $3N/4$. Plot these paths as in problem 3. Compute the shortest path 10 times and return the average length shortest path in both cases.

For more information on small world graphs, consult the paper

[1] Watts, D.J. and Strogatz, S.H. “Collective dynamics of ’small-world’ networks.” *Nature* **393** (6684) pp 40910. 1998