NAVAL POSTGRADUATE SCHOOL
Systems Engineering Department

SE4960: Network Concepts in Systems Engineering

T.H. Chung
Fall 2013

Homework #1

Issued: 03 Oct 12
Due: 10 Oct 12

Note: In the upper left hand corner of the *second* page of your homework set, please put the number of hours that you spent on this homework set.

1. (*Elements of graph theory*) Create a MATLAB function that will compute and return general graph properties for a given input graph represented as its adjacency matrix, `A`. The definition of your function should adhere to the following specification:

   `[D, L, adj_list, d_bar, diam] = graphSpecs_`*`Lastname`*`( A )`

   Your function should return the following properties of the graph:

   (a) `D`, the degree vector containing the node degrees
   (b) `adj_list`, the adjacency list representation of `A`
   (c) `L`, the graph Laplacian matrix
   (d) `d_bar`, the average degree over all nodes
   (e) `diam`, the diameter of the graph

   You can use the provided tester script, `graphSpecsTester.m`, to determine interoperability of your function.

   Deliverables: Submit your MATLAB function file using Sakai's Assignment page with the following naming convention, `graphSpecs_`*`Lastname`*`.m`.

(over⇝)

2. (*Shortest path algorithm*) Construct a MATLAB function that, for a given network, computes the shortest path from a start node to destination node and returns the sequence of nodes defining this shortest path and its length. Your function should have the following form:

```
[path, path_length] = shortestPath_Lastname( A, start, dest )
```

and should return the shortest path $\mathcal{P} = \{n_{\text{start}}, n_1, \ldots, n_{\text{dest}}\}$ as a vector `path`, given the starting node index, `start`, and destination node index, `dest`, according to the node indices of the input adjacency matrix, `A`.

You may use any formulation for computing the shortest path; however, the breadth-first search and labeling algorithm is a good place to start. Your function will be for accuracy in reporting the correct path (among the set of possibly multiple optimal paths) and length.

Deliverables: Submit your MATLAB function file using Sakai's Assignment page with the following naming convention, `shortestPath_Lastname.m`.

3. (*Random graphs*) Random graphs are characterized by the (independent) probability that an edge between node $i \in \mathcal{V}$ and node $j \in \mathcal{V}$ is present in the graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

   (a) Construct a MATLAB script which generates a (directed) random graph (in the sense of Erdös-Rényi) for given inputs of the number of nodes, $|\mathcal{V}|$, and the edge probability, p_{ij}. The computation should generate the adjacency matrix, A, of your random graph.

   (b) Using your function from Part, `graphSpecs_Lastname.m`, determine the distribution of degrees over all nodes, and plot the degree distribution as a histogram. (Hint: Use the `hist()` and/or `bar()` commands in MATLAB.) Be sure to provide appropriate axis labels and title containing information on number of nodes and edge probability.

   (c) Use the graph visualization tool, *Pajek*[1], to visualize your random graph. You may use any number of open-source scripts to convert your graph in MATLAB into the *Pajek*-compatible network file; I recommend the `write_matrix_to_pajek.m`[2] (available on Sakai site).

   Deliverables: Submit the following items using Sakai's Assignment page

   (a) your MATLAB script file with the following naming convention, `randomGraph_Lastname.m`;

   (b) a screen shot of the degree distribution plot from your MATLAB script;

   (c) a screen shot of (an artistic rendering of) your random graph output from *Pajek*;

   (d) any supporting *Pajek* `.net` file and/or project files used to create your visualization.

[1] *Pajek* wiki: `http://pajek.imfm.si/doku.php?id=pajek`

[2] From ModelGUI Wiki site: `http://mgui.wikidot.com/user-tutorial-matlab-pajek`