MATH 3353-001/2, Fall 2016
Lab # 6: **DUE 11/30/16**

To receive credit for this lab, you must turn in your diary that demonstrates you running these commands. Please name it as follows: `<Your_last_name>_lab6_diary.txt` (for example, `Barreiro_lab6_diary.txt`). There will be a place on Canvas for you to submit your file. I recommend that you save your commands in a `.m` file (for example, `myscript.m`): this will allow you to work out any mistakes before you run it and save your diary. Then run:

```
>> diary Barreiro_lab6_diary.txt
>> myscript
>> diary off
```

and you're done! **Don't forget to erase your diary before you do your submission run**

Introduction: In this lab, we will experiment with Google's *page rank* algorithm. The main idea is to rank pages based on how often they would be accessed by someone randomly wandering around the web clicking on links (something that might be done, literally, by a *web crawler*).

We can model the behavior of a web crawler through a Markov Chain: each web page is a state. When the crawler arrives at a page, it has an equal probability of following any of the links on that page. Thus

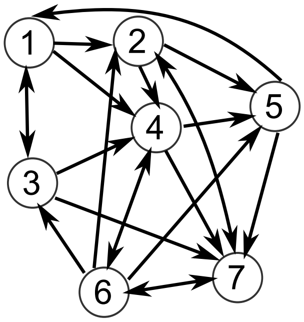
$$\mathbf{x}_{k+1} \quad = \quad P\mathbf{x}_k, \qquad \text{for } k = 0, 1, 2, \ldots \tag{1}$$

where P is an $N \times N$ matrix in which P_{ij} is the probability that the crawler will next visit state i, if it is currently at state j. How do we get P? You first create an *adjacency matrix* A, which is a matrix of ones and zeros with the rule that $A_{ij} = 1$ if site j has a link to site i; otherwise $A_{ij} = 0$. Then, scale each column of A so that the sum of each column is 1. The result is your stochastic matrix P.

1. [**Define a mini-web**] We begin by defining a "mini-web" to experiment with. Assume it has 7 web sites. Create both an adjacency matrix A and stochastic matrix P such that:

   - Site 1 links to sites 2,3, and 4.
   - Site 2 links to sites 4,5 and 7.
   - Site 3 links to sites 1,4 and 7.
   - Site 4 links to sites 5,6 and 7.
   - Site 5 links to site 1 and 7.
   - Site 6 links to sites 2,3,4,5, and 7.
   - Site 7 links to site 2 and 6.

   See figure (right). Print both A and P to your diary.



2. [**Crawl the web**] Suppose that a large number of crawlers started at site 1: i.e. $\mathbf{x}_0 = (1, 0, 0, 0, 0, 0, 0)$. Where would they be at $t = 1$, $t = 2$ (i.e. $\mathbf{x}_1$ and $\mathbf{x}_2$)? Print both to your diary.

3. [**Crawl the web, continued.**] Print the locations of the crawlers, as vectors, at times 5, 10, 20, and 50; that is, print $\mathbf{x}_5$, $\mathbf{x}_{10}$, $\mathbf{x}_{20}$, and $\mathbf{x}_{50}$.(Suggestion: use a `for` loop, as in Lab # 5).

4. [**Page rank**] The steady-state of this Markov chain — i.e., a vector $\mathbf{q}$ such that $P\mathbf{q} = \mathbf{q}$ — defines the *page rank* vector. Find $\mathbf{q}$ using the `eig` command as follows:

   (a) First, compute the eigenvalues and eigenvectors.
   ```
   >> [V,D]=eig(P);
   ```

   (b) The matrices `V` and `D` provide a diagonalization for P. That is, $PV = VD$. Use this fact to find $\mathbf{q}$: print $\mathbf{q}$ to your diary.

   (c) Each entry in $\mathbf{q}$ represents the page rank of a site. For example, $\mathbf{q}(1)$ is the page rank of site 1; $\mathbf{q}(6)$ is the page rank of site 6. Which site has the highest page rank? Which has the lowest? Print both integers to your diary.

5. [**How do you get a high page rank?**] Having a high page rank is good, because it means visitors will be more likely to stumble onto your site (and, your site will show up earlier in Google searches!). What characteristics might lead to a high page rank? Here are some ideas:

   (a) The *in-degree* of a node is the number of incoming connections. Find the in-degree of the 7 sites, as a vector $\mathbf{d}_{in}$, by performing an appropriate operation on the adjacency matrix A. Print $\mathbf{d}_{in}$ to your diary.

   (b) The *out-degree* of a node is the number of outgoing connections. Find the out-degree of the 7 sites, as a vector $\mathbf{d}_{out}$, by performing an appropriate operation on the adjacency matrix A. Print $\mathbf{d}_{out}$ to your diary.

   (c) Now, sort the page ranks from highest to lowest. Rearrange $\mathbf{d}_{in}$ and $\mathbf{d}_{out}$ in the same way. You can do this by hand, or using the `sort` command as follows:

   ```
   >>  [qsorted, Ivec]=sort(q, ’descend’);     % Ivec retains the original locations
                                               %    of the entries in qsorted.
   >>  din_sorted = din(Ivec);       % Use Ivec to rearrange din, dout in
                                     %    the same order.
   >>  dout_sorted = dout(Ivec);
   ```

   Print all three sorted vectors to your diary.

6. [**Getting a high page rank**] Choose the site with the lowest page rank. Suppose this was your site. *If you could change one connection to improve your page rank, what would it be?* Make a guess based on your findings in the previous question. Would you link to a popular site? Or is there someone you would convince to link to you? Or would you do something else entirely?

   Add a single link to A at a location of your choice, creating a new matrix B and stochastic matrix Q. Print both B and Q to your diary. Find the steady-state vector of Q, and print it to your diary. Now, report the new rank of **your site** to your diary. Did it improve?