Sections 4.4 and 4.5 Worksheet

1. Open the Matlab program StationaryDist, in the Files folder of Canvas. As written, the program gives the stationary distribution of the transition matrix

$$p = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 7/10 & 3/10 & 0 \\ 1 & 0 & 0 \end{bmatrix},$$

which was an example in class. Now, let's consider a queuing problem. A discrete time queuing system of capacity n consists of the person being served and those waiting to be served. The queue length X_n is observed each second. If $0 < X_n < N$, then with probability p, the queue size is increased by one by an arrival and, independently with probability r, it is decreased by one because the person being served finishes service. If $X_n = 0$, only an arrival is possible. Form a Markov chain with states given by the number of customers in the queue.

2. Write a program to simulate the queue in the previous problem.

   (a) You can simulate a queueing problem by iterating time. Create a 'for' loop like this:

```
p = 0.2;
r = 0.4;
tmax = 1000;
Xn = NaN(size(tmax));
for t = 1:tmax
(Draw random numbers x and y using the Matlab function rand)
(With probability x<p, someone arrives)
(With probability y<r, someone leaves)
end
```

   Here, p is the probability of an arrival, $tmax$ is the maximum number of seconds to run the simulation for, which you can adjust. Then we create a vector X_n to keep track of the queue length for each second we run the simulation for (NaN stands for 'Not a Number').

   (b) Why do we use the probability $x < p$ instead of $x > p$?

   (c) Modify the program above to keep track of the proportion of the time that the queue length is j, for $j = 0, 1, \ldots, n$, and the average queue length.

3. Consider the general Ehrenfest chain, i.e. the chain on the previous worksheet for general N.

   (a) Is the chain irreducible?

   (b) What is the period of each state?

   (c) Prove that $\pi(k) = \binom{N}{k}2^{-N}$ is a stationary distribution for this Markov chain

Real World Example: In order for search engines to be effective, they need to be able to rank the websites that searches return so that they can list the ones you might be most interested in first. On way to do this is using the PageRank algorithm. To implement this algorithm, we create a Markov chain where the states are all the websites. The transition probability p(website 1,website 2) is the probability that if you randomly clicked a link on website 1, it would take you to website 2. If this Markov chain has a unique stationary distribution, then that distribution is the percentage of time we would spend at each website if we are clicking links at random. This is a rough estimate of the importance of a website and can be used to sort the websites. Search engines such as Google use modified PageRank algorithm as part of their sorting methods.

For more information:

- "How Google works: Markov chains and eigenvalues" on the Klein Project Blog.
- "Application of Markov Chain in the PageRank Algorithm," Curtin University Technology, Science and Engineering Conference Paper.