Markov Chains Workshop **CS 480**

Write your answers below. When you are finished, turn this page in, but also share your Python code with me. If you are using Google Colab, share your code with: `lins.brian@gmail.com`

1. **Gambler's ruin problem.** Two gamblers are playing a game. Player 1 wins with probability p and player 2 wins the rest of the time (so with probability $1-p$). Each player starts with \$5, and whoever wins the game takes \$1 from the loser. They keep playing repeatedly until one player runs out of money.

   (a) Draw and label a graph for this Markov chain.

   (b) What is the transition matrix for this Markov chain?

   (c) If the probability that player 1 wins the game is $p = 60\%$, then how likely is player 1 to get all of the money in the end? Hint: The hardest part of this problem is entering the transition matrix into Python since it is an 11-by-11 matrix. You can use the numpy function `np.zeros((11,11))` to create an array where every entry is zero. Then you can change an entry in row i and column j by doing the following:

   ```
   import numpy as np
   Q = np.matrix(np.zeros((11,11)))
   i, j = 1, 2
   Q[i,j] = 0.6
   ```

   I recommend using a for-loop to set the entries of the transition matrix Q.

   (d) What if player 1 wins the game 80% of the time? Then how likely is player 1 to take all of the money?

2. **Coupon collector's problem.** A store has 20 different kinds of special coupons. A collector wants to collect at least one of each type of coupon. Every time the collector goes to the store, he gets one coupon which is equally likely to be any of the twenty types.

   (a) Draw and label the graph for this Markov chain. You don't have to draw the whole graph, but draw enough of it so that the pattern is clear. Then you can use ellipsis (...) for the middle part. Hint: When the collector has no coupons, there is a 100% chance he will get a new type of coupon when he goes to the store. But as he collects more types of coupons, the chance of getting a new type will get lower.

   (b) What is the minimum number of times the collector needs to go to the store in order to have at least a 50% chance of having every type of coupon, assuming he starts with none? Give your answer and explain your reasoning. Hint: When you calculate vQ^k, in Python, you can access the entry in row i, column j by using the command `(vQ**k)[i,j]`.

   (c) How many times would the collector have to go to the store to have a 90% chance of having all 20 types of coupon?