Math 135B, Winter 2023.

Homework 7

Due: Mar. 3, 2023

1. Show that the Markov chain in Problem 1 of Homework 4 has a unique invariant distribution and compute it. If, initially, both balls are in urn 1, approximate the probability that one of the urns is empty at time 1000.

2. Consider the Markov chain in Problem 2 of Homework 4.

(a) Show that it has a unique invariant distribution and compute it. What does the matrix P^n look like for large n?

(b) What proportion of time is the state of this chain strictly smaller than the previous state?

3. Consider the Markov chain in Problem 3 of Homework 4.

(a) Show that it has a unique invariant distribution and compute it.

(b) Compute the long-term proportion of tosses on which Alice uses coin 2.

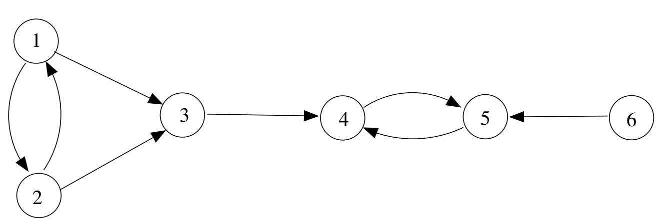
(c) Compute the long-term proportion of tosses on which Alice tosses Heads.

(d) Compute the long-term proportion of tosses on which Alice tosses Heads with coin 2.

(e) Bob pays Alice \$1 every time she tosses Heads with coin 1 and \$2 every time she tosses Heads with coin 2. Determine Alice's long-term average winnings per toss.

4. (A Wall Street job interview question.) You roll a fair six-sided die repeatedly and sum the numbers you roll. What is the expected number of rolls until the sum is a nonzero multiple of 4 for the first time?

5. Currently, it is not public knowledge how Google ranks the web pages in its search algorithm. The initial approach by S. Brin and L. Page in the mid 1990s, known as the *PageRank* algorithm, was using Markov chains for this purpose. Form an oriented graph, with N vertices, which are the web pages, in which an oriented edge goes from a web page x to a web page y if x contains a hyperlink to y. This creates a graph as in the following example with $N = 6$.



A random surfer begins at some page and then at every step chooses one of the outgoing arrows from the current page uniformly at random and then follows it to the next page; if there are no such such arrows, the surfer chooses one of the N pages uniformly at random and moves there. Let W be the transition matrix for the resulting Markov chain.

(a) Write down the matrix W for this example. Determine the classes and their recurrence or transience.

We see that the immediate problem is that the chain may not be irreducible. To deal with this problem, we introduce "damping." Pick a small $\epsilon > 0$. Then, the surfer follows the above algorithm with probability $1 - \epsilon$. Otherwise, with probability ϵ, it chooses a random state, i.e., evolves for one step according to the $N \times N$ transition matrix R that has all entries equal to $1/N$. A typical choice is $\epsilon = 0.15$; assume this for the rest of the problem. This gives the transition matrix

$$P = (1 - \epsilon)W + \epsilon R.$$

Using this transition matrix, the surfer's position has a unique invariant distribution, which is used to rank pages: the more often, in the long run, the surfer visits a web page x, the higher the rank of x.

(b) Rank the pages for the given example using this algorithm.

One problem with this version of *PageRank* is its vulnerability to the *Sybil attacks*. (The name comes from a famous, but possibly fictional, psychiatric case of a woman with multiple personality disorder.) That is, the owner of a page x can improve its rank by creating a number of new web pages $x_1, \ldots, x_k$, with links $x \to x_i$ and $x_i \to x$, for all $i = 1, \ldots, k$. In our example, say, the owner of page 5 creates a single new page 7 (so $k = 1$) with links $5 \to 7$ and $7 \to 5$ (but no other new links).

(c) Using *PageRank*, determine the new ranking of pages after the described Sybil attack in our example.