Assignment 2

Lilian Kourti
CME 241

<u>Problem 1</u>

The board of Snakes and Ladders consists of 100 blocks, numbered from 1 to 100. The starting position is defined to be off the board. Let's call that block 0. Hence, the state space is defined as follows:

$$S = \{0, 1, \dots, 99, 100\}$$

where each state corresponds to a block on the board (or the off board starting position). The starting state is 0 and the terminating state is 100 ($T = \{100\}$).

Since there exist a couple different variations of the game, it is underlined that in order to win the player will need to roll the exact number that will land them on the last square (100). If the player rolls a higher number than needed to land exactly on 100, their piece does not move and remains there until their next turn, when they can roll again.

The transition matrix P holds the transition probabilities and $P[i, j]$ is the probability of transitioning from state i to state j, $\forall i \in \{0, \dots, 99\}$ and $\forall j \in \{0, \dots, 100\}$. Let's forget for a while that there are snakes and ladders, then the transition matrix would be the following:

$$P = \begin{bmatrix}
0 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 2/6 & 1/6 & 1/6 & 1/6 & 1/6 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 3/6 & 1/6 & 1/6 & 1/6 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 4/6 & 1/6 & 1/6 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 5/6 & 1/6
\end{bmatrix}$$

The transition matrix has the above form because by rolling the dice we can only move forward and make at most 6 steps with equal probability. The last rows of the transition matrix (corresponding to states 95-99) have a different pattern than the rest of the matrix because of the winning condition explained above.

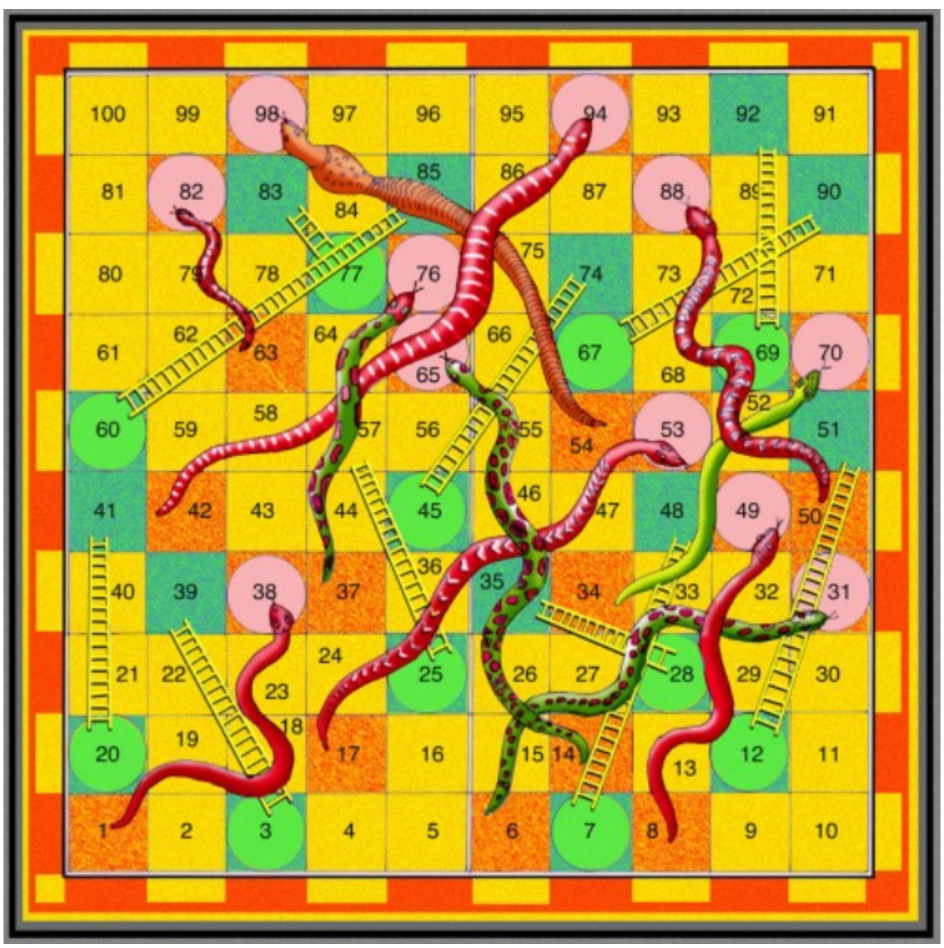
Now, let's add the snakes and the ladders and assume that we have a valid board. We define a mapping $M : S \to S$ that corresponds to the effect of the snakes and the ladders. For example, if there exists a snake (or ladder) from block i to block j with $i > j$ (similarly for ladders $i < j$) then we get the following mapping: $M[i] = j$. Therefore, the transition matrix changes and the transition probabilities become:

$$P[i, M[j]] += \begin{cases} 1/6, & i \in \{0, \dots, 94\}, \quad j \in \{i+1, \dots, i+6\} \\ 0, & i \in \{0, \dots, 94\}, \quad j \notin \{i+1, \dots, i+6\} \\ (6 - 100 + i)/6, & i \in \{95, \dots, 100\}, \quad j = i \\ 1/6, & i \in \{95, \dots, 100\}, \quad j \in \{i+1, \dots, 100\} \\ 0, & i \in \{95, \dots, 100\}, \quad j \notin \{i, \dots, 100\} \end{cases}$$

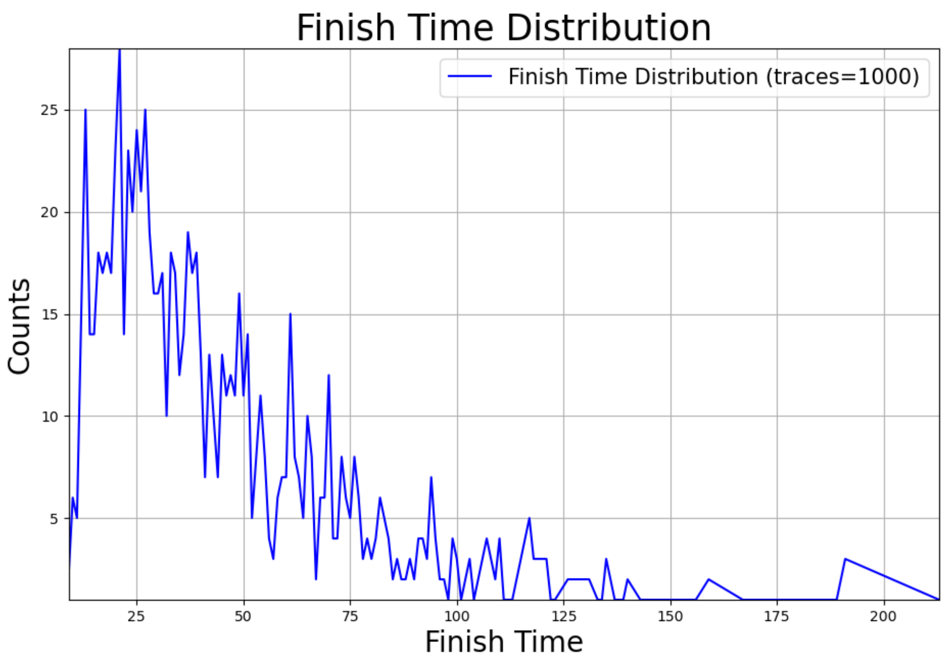
Assuming that we have initialized all the probabilities to zero.

<u>Problem 2</u>

[See RL-book/_lkourti/Assign2/prob2.py]
The code was tested using the following snakes and ladders structure:



After creating 1000 sample traces using the method traces, the probability distribution of time steps to finish the game is plotted:



Problem 3

- When the frog jumps it randomly chooses from the possible landing places.
- The frog keeps jumping forward until it crosses the river (reaches the other bank).
- What is the expected number of jumps for different river widths?

Let's consider a river with n landing places, where the first n-1 landing places correspond to waterlilies and the n-th landing places corresponds to the river bank where the frog want to go. To address this problem as a MP, we define the state space as:

$$S = \{0, 1, \dots, n-1, n\}$$

where state 0 corresponds to the starting bank ($S_0 = 0$), states $1 - n$ correspond to the waterlilies and state n corresponds to the finish bank ($T = \{n\}$).

The transition probability matrix P will have entries $P[i, j]$ equal to the probability of transitioning from state i to state j, $\forall i \in \{0, \dots, n-1\}$ and $\forall j \in \{0, \dots, n\}$.

$$\mathbf{P} = \begin{bmatrix} 0 & 1/n & 1/n & \cdots & 1/n & 1/n & 1/n \\ 0 & 0 & 1/(n-1) & \cdots & 1/(n-1) & 1/(n-1) & 1/(n-1) \\ 0 & 0 & 0 & \cdots & 1/(n-2) & 1/(n-2) & 1/(n-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix}.$$

The transition matrix has the above form because if the frog is at the $(k-1)$-th landing place and has k landing places in front of him, he chooses with probability $1/k$ to jump on one of them.

Let $\mathbf{t}$ be the vector of expected number of jumps to reach the terminal state n. More precisely, $\forall i = 0, \ldots, n-1$ $\mathbf{t}[i]$ is the expected number of jumps to go from state i to the terminal state n. Therefore, we are interested in finding $\mathbf{t}[0]$ since the frog always starts from state 0. The following is also helpful in finding $\mathbf{t}[0]$:

$$\mathbf{t}[j] = 1 + \sum_{k=0}^{n-1} P[j,k]\mathbf{t}[k], \quad \forall j = 0, \ldots, n-1$$

In matrix notation:

$$\mathbf{t} = \mathbf{1} + \mathbf{P}_{NT}\mathbf{t}$$

where $\mathbf{P}_{NT}$ is the transition matrix without the column corresponding to the terminal state n, since $\mathbf{t}[n] = 0$. Then all we need to do is solve the following system of equations for $\mathbf{t}$:

$$(\mathbf{I} - \mathbf{P}_{NT})\mathbf{t} = \mathbf{1}$$

Hence, we found that the expected number of jumps is $\mathbf{t}[0]$. The *Frog Puzzle* is solved for a general n in RL-book/_lkourti/Assign2/prob3.py. For example, for $n = 10$, the expected number of jumps is 2.93, and for n = 20, the expected number of jumps is 3.6.

Problem 4

For this problem, the expected number of dice rolls to finish the game of Snakes and Ladders will be calculated as the expected return. To do so, the game will be modeled as Finite Markov Reward Process, where each step will yield a reward of 1 and there won't be any discounting ($\gamma = 1$). Since we know that in finite number of steps the game will finish there is no risk by omitting discounting. This way the expected number of dice rolls to finish the game and the expected return are equivalent and can be calculated through the Value Function. See RL-book/_lkourti/Assign2/prob4.py. Using the get_value_function_vec method of FiniteMarkovRewardProcess, we get that the expected number of dice rolls to finish the game is 48.97818.

Problem 5

Since we are dealing with an infinite-states, non-terminating MRP, we will generate several simulation traces of specified length and approximate the Value Function (expected return) for a given starting price as the average return over these traces. Keeping $\gamma < 1$ is essential to prevent the sum from blowing up. See RL-book/_lkourti/Assign2/prob5.py for the implementation.