CS 7642: Reinforcement Learning and Decision Making **Homework #1**

Planning in MDPs

1 Problem

1.1 Description

You are given an N-sided die, along with a corresponding Boolean mask vector, `is_bad_side` (i.e., a vector of ones and zeros). You can assume that $1 < N \leq 30$, and the vector `is_bad_side` is also of size N and 1 indexed (since there is no 0 side on the die). The game of DieN is played as follows:

1. You start with 0 dollars.
2. At any time you have the option to roll the die or to quit the game.
   (a) **ROLL**:
      i. If you roll a number not in `is_bad_side`, you receive that many dollars (e.g., if you roll the number 2 and 2 is not a bad side – meaning the second element of the vector `is_bad_side` is 0, then you receive 2 dollars).

         Repeat step 2.

      ii. If you roll a number in `is_bad_side`, then you lose all the money obtained in previous rolls and the game ends.
   (b) **QUIT**:
      i. You keep all the money gained from previous rolls and the game ends.

1.2 Procedure

- The answer you provide should be the number of dollars you expect to win for a specific configuration of the die, if you follow an optimal policy. That is, what is the value of the optimal state-value function for the initial state of the game (starting with 0 dollars)?

  Provide answers for the specific problems you are given on Canvas. Your answer must be correct to 3 decimal places, truncated (e.g., 3.14159265 becomes 3.141).
- To solve this problem, you will need to determine an optimal policy for the game of DieN, given a particular configuration of the die. As you will see, the action that is optimal will depend on your current bankroll (i.e., how much money you've won so far).
- You can try solving this problem by creating an MDP of the game (states, actions, transition function, reward function, and assume a discount rate of $\gamma = 1$) and then calculating the optimal state-value function.

2 Examples

The following examples can be used to verify that your agent is implemented correctly.

- Input : `is_bad_side` $= [1, 1, 1, 0, 0, 0]$, Output : 2.583
- Input : `is_bad_side` $= [1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0]$, Output : 7.379
- Input : `is_bad_side` $= [1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0]$, Output : 6.314

3 Resources

The concepts explored in this homework are covered by:

3.1 Lectures

- Lesson 1: Smoov & Curly's Bogus Journey

3.2 Readings

- Chapter 3 (3.6 Optimal Policies and Optimal Value Functions) and Chapter 4 (4.3-4.4 Policy Iteration, Value Iteration) of Sutton and Barto 2020
- Chapters 1-2 of Littman 1996

4 Submission Details

The due date is indicated on the Canvas page for this assignment.
Make sure you have set your timezone in Canvas to ensure the deadline is accurate.
Submit your answers on Canvas, as outlined in section 1.2. You will have a total of 10 submission attempts - only the highest score is kept.

References

[Lit96] Michael Lederman Littman. *Algorithms for Sequential Decision Making*. 1996.

[SB20] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. 2nd Ed. MIT press, 2020. URL: http://incompleteideas.net/book/the-book-2nd.html.