Homework 3: Dynamic Programming
Due: Monday, September 28th 11:59 pm

The purpose of this project is to study different properties of dynamic programming methods.

Problem

A gambler has the opportunity to make bets on the outcomes of a sequence of coin flips. If the coin comes up heads, he wins as many dollars as he has staked on that flip; if it is tails, he loses his stake. The game ends when the gambler wins by reaching his goal of $100, or loses by running out of money.
On each flip, the gambler must decide what portion of his capital to stake, in integer numbers of dollars. This problem can be formulated as an **undiscounted** ($\gamma = 1$), **episodic**, finite MDP. The state is the gambler's capital $s \in \{1,2,\dots,99\}$ and the actions are stakes $a \in \{0,1,\dots,\min(s, 100-s)\}$. The reward is zero on all transitions except those on which the gambler reaches his goal, when it is +1.
The state-value function then gives the probability of winning from each state. A policy is mapping from levels of capital to stakes. The optimal policy maximizes the probability of reaching the goal.
Let's p_h denote the probability of the coin coming up heads. If p_h is known the problem can be solved using value iteration.

Part I

Implement the Gambler's problem and then implement **value iteration** to solve the MDP for three scenarios where $p_h = \{0.4, 0.25, 0.55\}$ and find the optimal value function and optimal policy.
Tip: When implementing, you might find it convenient to introduce two dummy states corresponding to termination with capital of 0 and 100, giving them values of 0 and 1 respectively.

For all three scenarios:
1. Plot the change in the value function over successive sweeps of value iteration w.r.t capital (state).
2. Plot the final policy w.r.t capital (state).

Part II

Answer the following questions:
1. What action does your optimal policy suggest for capital of 50? What about for capital of 51?
2. Why do you think your optimal policy is a good policy? Explain.

Part III*

Test the algorithm by decreasing θ the threshold for accuracy of value function estimation. What happens when $\theta \to 0$.

Evaluation: we will grade your submission according to the following table:

Item	**COMP4600**	**COMP5300**
Implementation of the problem (states, actions, transition, rewards)	20	20
Implementation of value iteration (value and policy)	40	30
Plotting value and optimal policy	30	30
Answering questions in Part II	10	10
Answering questions in Part III *	-	10

Note 1: The parts marked with * are optional for COMP4600 (undergraduates) but mandatory for COMP5300 (graduates).

Note 2: We do not accept code in any programming language other than Python 3 (do not use Python 2).

Note 3: For the implementation of the testbed and the algorithms, you are not allowed to rely on any python library other than `numpy` and `matplotlib` (for plotting).

Note 4: All the code, plots, explanations should be included in a single Jupyter Notebook (`.ipynb`) file. Include your name as part of the filename and submit through Blackboard.

Submission: By 11:59pm on Monday, September 28th 2020, submit your `student_name.ipynb` files on Blackboard. Make sure everything is entirely contained within this file and it runs without any error.

Late Policy: Up to two late days are allowed, but a grade penalty of 50% and 75% will be applied at the first and second day, respectively.