Department of Electrical and Computer Engineering National University of Singapore Dr Fan Shi, Dr Armin Lederer	Robotics and Embodied Artificial Intelligence CEG5306 Assignment 2: model-based and model-free tabular RL	Autumn 2025

1. **Upper Confidence Bounds in Model-based RL (∼ 7 pts)**
   The principle of optimism in the face of uncertainty can be realized using upper confidence bounds as demonstrated for multi-armed bandits. This paradigm can also be used for model-based reinforcement learning. For simplicity, assume an oracle providing you with a learning error bound for estimated transition probabilities in the following.

   (a) A core element for implementing model-based reinforcement learning using upper confidence bounds is an approach to optimistically plan in MDPs. Describe how you can perform optimistic planning based on value iteration (pseudocode, mathematical equations, text).

   (b) Write pseudocode for an upper confidence bound algorithm in model-based reinforcement learning.

2. **Monte Carlo Control (∼ 5 pts)**
   Consider an unknown MDP with 3 ($\{0, 1, 2, \}$) states and 2 actions ($\{0, 1\}$). We consider a known reward function

$$r(s, a) = \begin{cases} 0 & \text{if } s = 0 \\ 1 & \text{if } s = 1 \\ 10 & \text{if } s = 2. \end{cases} \tag{1}$$

   The following trajectories have been observed so far:

trajectory idx	state	action	state	action	state
1	0	0	1	0	1
2	0	1	2	1	0
3	0	0	0	0	1

   (a) Approximate the state-action value function Q using Monte Carlo estimation. Assign a value of 0 for all state-action pairs that are not observed.

   (b) After several more roll-outs with the same policy, we obtain the following estimate for the state-action value function

$$\begin{aligned} Q(0, 0) &= 50 & Q(0, 1) &= 47 \quad (2) \\ Q(1, 0) &= 50 & Q(1, 1) &= 49 \quad (3) \\ Q(2, 0) &= 55 & Q(2, 1) &= 50. \quad (4) \end{aligned}$$

   Specify the probability distribution of the ϵ-greedy policy induced by this Q-function for $\epsilon = 0.15$.

3. **Q-learning (∼ 7 pts)**
   Consider an unknown MDP with 3 ($\{0, 1, 2, \}$) states and 2 actions ($\{0, 1\}$). We consider a known reward function

$$r(s, a) = \begin{cases} 0 & \text{if } s = 0 \\ 1 & \text{if } s = 1 \\ 10 & \text{if } s = 2. \end{cases} \tag{5}$$

   The following trajectories have been observed so far:

trajectory idx	state	action	state	action	state
1	0	0	1	0	1
2	0	1	0	1	2
3	0	0	1	1	0

(a) Estimate the function Q^* for $\gamma = 0.9$ like the Q-learning algorithm presented in lecture 3 does. Assume that the function Q^* is initialized with 0 for all states and actions and use a learning rate of $\alpha = 1$.

(b) Assume that after several more episodes, you obtain the following estimate of the function Q^*:

$$Q^*(0,0) = 45 \qquad Q^*(0,1) = 55 \tag{6}$$
$$Q^*(1,0) = 50 \qquad Q^*(1,1) = 49 \tag{7}$$
$$Q^*(2,0) = 50 \qquad Q^*(2,1) = 60. \tag{8}$$

Compute the deterministic policy maximizing the function Q^*.

(c) As shown in the lecture, we can use gradient descent techniques to enable Q-learning with function approximation. Derive the update step when using linear function approximation for the value function, i.e., $Q(s, a; \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \boldsymbol{\phi}(s, a)$ where $\boldsymbol{\theta}$ is a parameter vector and $\boldsymbol{\phi}(\cdot, \cdot)$ are known nonlinear features. Express the update step in terms of the Q-function and reward function.