Assignment 3.1: The Agent's Mind

MDPs, Bellman Equations, and the Nature of Reward

Objective

This assignment contains no Python code. It is a "Pen and Paper" exercise. Reinforcement Learning is mathematically grounded. If you cannot solve these simple environments by hand, you will not deeply understand how Monte Carlo control works.

Question 1: The "Cliff Walker" (Manual Calculation)

Consider a tiny 1D GridWorld with 4 states: S_0, S_1, S_2, S_{Term}.

- **Transitions:** From any state S_i, you can move **Right** (S_{i+1}) or **Left** (S_{i-1}).
- **Boundaries:** Moving Left from S_0 keeps you in S_0. Moving Right from S_2 takes you to S_{Term} (Game Over).
- **Rewards:**
  - Transition to S_{Term}: $+\mathbf{10}$
  - Any other transition (Left or Right): **-1** (Living penalty)
- **Discount Factor (γ):** 0.9

The Tasks:

1. **Calculate Return (G_t):** An agent starts at S_1. It takes the path:
   $S_1 \xrightarrow{Right} S_2 \xrightarrow{Left} S_1 \xrightarrow{Right} S_2 \xrightarrow{Right} S_{Term}$.
   Calculate the total discounted return G_0 for this episode. Show your working using powers of γ.
2. **Value Function (v_π):** Consider a "Random Drunk" policy π where the agent chooses Left or Right with probability 0.5. Write down the **Bellman Expectation Equation** for the value of state S_2 (i.e., $v_\pi(S_2)$).
   Hint: Express $v_\pi(S_2)$ in terms of the immediate rewards and the values of its neighbors ($v_\pi(S_1)$ and $v_\pi(S_{Term})$).

Question 2: The Philosophy of Reward (Design)

One of the hardest parts of RL is designing the reward function. A poorly designed reward leads to "Reward Hacking."

Scenario: You are training a robot to clean a messy room.

- **State:** Camera image of the room.
- **Action:** Move, Suck, Idle.
- **Your Reward Design:** You give the agent a reward of $+\mathbf{1}$ every time its sensors detect that "Dust" has been "Sucked Up."

The "What If":

The agent eventually learns a policy that maximizes total reward, but the room never stays clean. In fact, the room gets dirtier over time. **Explain exactly what behavior the agent has likely learned to exploit your reward function.**
(Hint: Think about where the dust goes after it is sucked up).

Question 3: The Discount Factor (Concept)

The discount factor γ represents how much the agent cares about the future.

Part A: The Math

Why do we mathematically *need* $\gamma < 1$ for continuous (infinite horizon) tasks? What would happen to the value function $v_\pi(s)$ if the task never ends, rewards are always $+1$, and $\gamma = 1$?

Part B: The Intuition

Imagine a generic MDP.

- **Case 1:** $\gamma = 0$.
- **Case 2:** $\gamma = 0.99$.

Explain in plain English how the behavior of the agent differs in these two cases. Which agent is "Impulsive" and which is "Strategic"?

Question 4: The Brain Teaser (Nuance)

This question tests your understanding of the **Reward Hypothesis**.

Suppose we have an environment where the agent's goal is to reach a Goal State in the shortest number of steps.

- **Original Setup:** Every step gives a reward of $R = -1$. Reaching the goal ends the episode. $\gamma = 1$. The optimal policy finds the shortest path.

The Modification: A developer decides to "boost" the agent's morale by adding a constant $C = +2$ to every single reward.
Now, every step gives a reward of $R_{new} = (-1 + 2) = +1$.

The Question:

Does the optimal policy π_* change?

- If yes, describe what the new optimal agent will do.
- If no, explain why.

Hint: Think about what the agent wants to maximize. Does it want to finish the game, or keep collecting rewards?

Submission Instructions

- Submit a PDF with your answers.
- Handwritten (scanned) or LaTeX submissions are both accepted.
- For Q4, a single sentence answer is not enough. You must explain the logic.