說明：請各位使用此template進行Report撰寫，如果想要用其他排版模式也請註明題號以及題目內容（請勿擅自更改題號），最後上傳前，請務必轉成PDF檔，並且命名為report.pdf，否則將不予計分。

學號：R12945060　　系級：生醫電資所　　姓名：羅佳蓉

1. (0.5%) Please write down the Bellman consistency equation in terms of V^{π} on both sides.

$$V^{\pi}(s) = \sum_{a \in A} \pi(s,a) \sum_{s' \in S} p(s'|s,a) \left[R(s,a) + \gamma V^{\pi}(s') \right]$$

- $V^{\pi}(s)$: The expected cumulative reward starting at state s and following policy π.
- $\pi(s,a)$: The probability of taking action a in state s under policy π.
- $p(s' \mid s,a)$: The probability of transitioning to state s´ from state]s by taking action a.
- $R(s,a)$: The immediate reward for taking action a in state s.
- γ: The discount factor, controlling the weight of future rewards.

2. (0.5%) Please implement the epsilon-greedy algorithm or the UCB algorithm. Paste the code and compare the public leaderboard scores of it and the default greedy algorithm (directly choose the state with maximum value).

- epsilon-greedy algorithm

```
def epsilon_greedy_action_selection(env, state, value_table, epsilon):
    actions = env._knight_moves(state[:2])
    if not actions:
        return state[:2]
    if random.random() < epsilon:
        return random.choice(actions)
    else:
        return greedy_action_selection(env, state, value_table)
```

epsilon-greedy score: 85.2

- UCB algorithm

```
def ucb_action_selection(env, state, value_table, count_table, c=2):
    actions = env._knight_moves(state[:2])
    if not actions:
        return state[:2]
    total_visits = sum(
        count_table[action[0], action[1], state[2], state[3]] for action in actions
    )
    ucb_values = [
        value_table[action[0], action[1], state[2], state[3]] +
        c * np.sqrt(np.log(total_visits + 1) /
                    (count_table[action[0], action[1], state[2], state[3]] + 1e-5))
        for action in actions
    ]
    return actions[np.argmax(ucb_values)]
```

UCB score: 87.8

3. (1%) How to encourage the agent to catch the pawn as soon as possible? Please make two modifications (for example, change the reward function, dis count factor, …)
   a. What is your first modification? How does it affect your public score?
      REWARD_STEP = -0.1
      Increase the penalty for each step taken by the agent. This forces the a gent
      to minimize the number of steps required to catch the pawn.

   b. What is your second modification? How does it affect your public score?
      GAMMA = 0.85
      Decrease the discount factor to reduce the weight of long-term rewards. This
      encourages the agent to prioritize immediate rewards, which aligns with catching the pawn quickly.

Modification	Public Score Before	Public Score After
Increase Step Penalty	85.5	91.2
Decrease Discount Factor	85.5	87