COMP 7404 Computational Intelligence and Machine Learning

The University of Hong Kong

Department of Computer Science

3035348102 ZHANG Yupeng

Question 1:

Value Iteration: in this question, I just use the formulas from the slide to compute these q values, and choose the maximum one as the action. The key point is the **Bellman equation**.

$$V_{k+1}(s) \leftarrow \max_{a} \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_k(s')]$$

Question 2:

Bridge Crossing Analysis: I only change the value of noise parameter to be 0, because I think the agent should act according to the policy. The noise parameter will make the agent go randomly.

Question 3:

Policies: in this question, I think negative reward would force the agent go to the near terminal states, while positive one may have inverse effect. As for discount and noise parameters, they are different compared to reward.

Question 4:

Q-Learning: this question is quite similar with the first question, I only change the iteration process to the **TD learning**.

$$sample = R(s,a,s') + \gamma \max_{a'} Q_k(s',a')$$

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\ sample$$

Question 5:

Epsilon Greedy: both the previous question and this question are part of Q-learning, and I just make the *getAction* method return a random action.

Question 6:

Bridge Crossing Revisited: I must say, only 50 episodes are too less to get a quite accurate outcome. Therefore, I just return '***NO POSSIBLE***'.

Question 7:
Q-Learning and Pacman: There's no need to write any piece of code.

Question 8:
Approximate Q-Learning: the instruction page has quite detailed guide, so I just finish it step by step.