- Suppose someone gives you a randomized policy π in a continuing reinforcement learning programming problem and claims it is optimal. How would you construct a deterministic policy π which is optimal?
  Justify your answer.
- Is it always true that $J_\pi(s) \leq \max_a Q_\pi(s, a)$?
- Suppose all the rewards in a continuing reinforcement learning problem are in [0,1] and we have $\gamma = 0.9$.
  As in the lecture notes, we use J^* to denote the rewards-to-go under the optimal policy.
  First, give an upper bound on $||J^*||_\infty$.
  Second, how many iterations does it take until we can guarantee that value iteration produces J_t with $||J_t - J^*||_\infty \leq 0.01$ starting from $J_0 = \mathbf{0}$?
- Look at Eq. (*) in the lecture notes on Q-learning. It was derived under the assumption that the policy π is deterministic. What should the corresponding equation be when the policy is randomized?

- Consider an MDP with two states, A and B. In A, there are two actions you can take. Action 1 keeps you in state A, with a reward of one. Action 2 moves you to B, with a reward of zero. In state B, there is only one action to take, which keeps you in B with a reward of 2.

  You want to use temporal difference learning to evaluate the rewards-to-go of the "choose at random" policy.
  You generate the following two sample paths:
  $s_1 = A, a_1 = 1, s_2 = A, a_2 = 2, s_3 = B$
  $s_1 = A, a_1 = 2, s_2 = B, a_2 = 1, s_3 = B, a_3 = 1, s_4 = B$
  Use temporal difference learning to come up with estimates of the rewards-to-go from both states starting from [16,16].

- Consider an MDP with two states, A and B. In A, there are two actions you can take. Action 1 keeps you in state A, with a reward of one. Action 2 moves you to B, with a reward of zero. In state B, there is only one action to take, which keeps you in B with a reward of 2.

  (i) Suppose you start from [16,16,16]. Starting at state A, write out the first three iterations of Q-learning with $\epsilon = 0$ starting from

  $s_1 = A, a_1 = 1, s_2 = A, a_2 = 2, s_3 = B, a_3 = 1$

  (ii) Write code to implement Q-learning to find the optimal Q-values in this example.