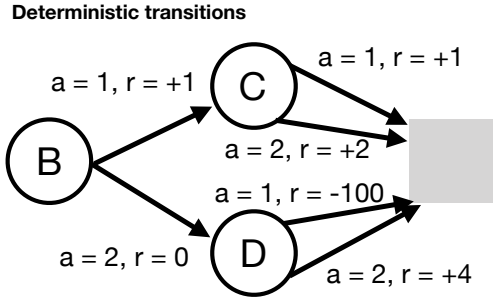
CMPUT 365: Introduction to Reinforcement Learning, Winter 2023
Worksheet #7: Temporal Difference Methods for Control

Manuscript version: *#d68e73-dirty* - 2025-03-06 14:20:03-07:00

Question 1. Consider an episodic MDP with the states $B, C,\ D$, and the terminal state T (i.e. $\mathcal{S} = \{B, C, D, T\}$), and 2 actions ($\mathcal{A} = \{1, 2\}$) with transitions and rewards as shown on the figure below.



Assume that the action values are initialized $Q(s, a) = 0$, for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$. The agent takes actions according to an ϵ-greedy policy with $\epsilon = 0.1$. Set the discount factor $\gamma = 1.0$.

1. Determine an optimal policy and the optimal action-value function.
2. In the remainder of the problem, we will consider the Sarsa and the Q-learning algorithms where the initial action-values are set to zero. Further, the stepsize is set to $\alpha = 0.1$. Consider the episodic data $S_0 = B, A_0 = 2, R_1 = 0, S_1 = D, A_1 = 2, R_2 = 4$. What are the action-values obtained by running Sarsa on this data for the various states?
3. What are the action-values obtained by running Q-learning on the above data for the various states?
4. Let's consider one more episode: $S_0 = B, A_0 = 2, R_1 = 0, S_1 = D, A_1 = 1, R_2 = -100$. What are the action-values obtained by running Sarsa on this data for the various states starting from the previous estimates obtained with Sarsa on the first episode's data? What are the action-values obtained by running Q-learning on this data for the various states starting from the previous estimates obtained with Q-learning on the first episode's data?
5. Assume the next episode's data is the same as in Part 4. Again, write down the action-value estimates after running Sarsa on this data, continuing from the previously obtained estimates. Do the same for Q-learning.
6. Do you notice any relationship between the values you obtained by emulating Sarsa and those that you got by emulating Q-learning? What is the relationship and why does it hold?

Solution.

1. From a visual inspection of the above MDP, clearly the optimal policy is $\pi^*(2|B) = 1$, $\pi^*(2|C) = 1$, $\pi^*(2|D) = 1$, and it assigns a probability of zero for all other state-action pairs. Next, recall that the Bellman optimality equation for the action value function is

$$q^*(s, a) = \sum_{s', r'} p(s', r'|s, a)[r' + \gamma \max_{a'} q^*(s', a')].$$

Using this, we obtain $q^*(C, 1) = 1, q^*(C, 2) = 2, q^*(D, 1) = -100, q^*(D, 2) = 4, q^*(B, 1) = 1 + q^*(C, 2) = 3$, and $q^*(B, 2) = 0 + q^*(D, 2) = 4$.

2. **SARSA:**

$$Q(B,2) = Q(B,2) + \alpha \cdot \big[r(B,2) + \gamma Q(D,2) - Q(B,2)\big] = 0 + 0.1\big[0 + 0 - 0\big] = 0,$$
$$Q(D,2) = Q(D,2) + \alpha \cdot \big[r(D,2) + \gamma Q(T,\cdot) - Q(D,2)\big] = 0 + 0.1\big[4 + 0 - 0\big] = 0.4.$$

3. Q**–learning:**

$$Q(B,2) = Q(B,2) + \alpha \cdot \big[r(B,2) + \gamma \max_{a'} Q(D,a') - Q(B,2)\big] = 0 + 0.1\big[0 + 0 - 0\big] = 0,$$
$$Q(D,2) = Q(D,2) + \alpha \cdot \big[r(D,2) + \gamma \max_{a'} Q(T,a') - Q(D,2)\big] = 0 + 0.1\big[4 + 0 - 0\big] = 0.4.$$

4. **SARSA:**

$$Q(B,2) = Q(B,2) + \alpha \cdot \big[r(B,2) + \gamma Q(D,1) - Q(B,2)\big] = 0 + 0.1\big[0 + 0 - 0\big] = 0,$$
$$Q(D,1) = Q(D,1) + \alpha \cdot \big[r(D,1) + \gamma Q(T,\cdot) - Q(D,1)\big] = 0 + 0.1\big[-100 + 0 - 0\big] = -10.$$

Q**–learning:**

$$Q(B,2) = Q(B,2) + \alpha \cdot \big[r(B,2) + \gamma \max_{a'} Q(D,a') - Q(B,2)\big] = 0 + 0.1\big[0 + 0.4 - 0\big] = 0.04,$$
$$Q(D,1) = Q(D,1) + \alpha \cdot \big[r(D,1) + \gamma \max_{a'} Q(T,a') - Q(D,1)\big] = 0 + 0.1\big[-100 + 0 - 0\big] = -10.$$

5. **SARSA:**

$$Q(B,2) = Q(B,2) + \alpha \cdot \big[r(B,2) + \gamma Q(D,1) - Q(B,2)\big] = 0 + 0.1\big[0 + -10 - 0\big] = -1,$$
$$Q(D,1) = Q(D,1) + \alpha \cdot \big[r(D,1) + \gamma Q(T,\cdot) - Q(D,1)\big] = -10 + 0.1\big[-100 + 0 + 10\big] = -19.$$

Q**–learning:**

$$Q(B,2) = Q(B,2) + \alpha \cdot \big[r(B,2) + \gamma \max_{a'} Q(D,a') - Q(B,2)\big] = 0.04 + 0.1\big[0 + 0.4 - 0.04\big] = 0.076,$$
$$Q(D,1) = Q(D,1) + \alpha \cdot \big[r(D,1) + \gamma \max_{a'} Q(T,a') - Q(D,1)\big] = -10 + 0.1\big[-100 + 0 + 10\big] = -19.$$

6. We observe that the SARSA and Q–learning updates are same for $Q(D,1)$, and they are different for $Q(B,2)$. For $Q(B,2)$, Q–learning makes the update using the next state action pair as $(D,2)$, even though the sampled next-state-action pair is $(D,1)$, because $Q(D,2)$ is higher then $Q(D,1)$.

□

Question 2. Answer the following:

1. Give at least two conditions that are "necessary" for Sarsa to produce a sequence $(Q_t)_{t\geq 0}$ of value function estimates that converges to q^* with probability one. For each condition, justify why is it "necessary", that is even if all the other conditions hold and the condition in consideration is violated, then convergence fails to hold.

2. Can any of these conditions be satisfied when Sarsa is used an in episodic MDP with exploring starts (and which one)? Why or why not?

Solution. 1. The required conditions are as follows

Condition 1: All the state-action pairs are visited infinitely often. As a specific example, consider an episodic MDP with a single state, two actions, both terminating (essentially a bandit problem). Assume that one of the actions, call this a_0, is chosen only once. Further, assume that the reward for this action is Bernoulli with parameter 0.5. Then, $Q_1(s,a_0) = Q_2(s,a_0) = \cdots$, and $Q_1(s,a_0) \in \{0,1\}$ (i.e. the action values stop updating after the first timestep) while $q^*(s,a_0) = 0.5$. Hence, convergence to q^* fails.

Condition 2: The action-selection mechanism is such that it selects greedy actions in the limit as $t \to \infty$. A simple example where this condition is violated when Sarsa is used while following a fixed memoryless policy π. (This setting is the prediction setting, as opposed to the control setting.) In this case, if Q_t converges, it can only converge to q_π.

Condition 3: The stepsizes need to satisfy the Robbins-Monro conditions: Allowing the stepsizes to depend on the state-action pairs, we need that both $\sum_{t\geq 0} \alpha_t(S_t,A_t) = \infty$ and $\sum_{t\geq 0} \alpha_t^2(S_t,A_t) < \infty$ hold with probability one, while $0 \leq \alpha_t(S_t,A_t)$ for all $t \geq 0$.

To see an example when convergence fails to hold, consider the bandit example again. In that example, Sarsa reduces to mean estimation of the reward for both the actions. Consider the case first when $\sum_{t\geq 0} \alpha_t(S_t,A_t) < \infty$. Say, $A_0 = a_0$ and $\alpha_0(s,a_0) = 1$ and $\alpha_t(s,a_0) = 0$ otherwise. Then, as in the first example, $Q_1(s,a_0) = Q_2(s,a_0) = \cdots$, and $Q_1(s,a_0) \in \{0,1\}$ while $q^*(s,a_0) = 0.5$. Hence, convergence to q^* fails. Now, for the case when $\sum_{t\geq 0} \alpha_t^2(S_t,A_t) = \infty$ assume that $\alpha_t(S_t,A_t) = \alpha > 0$. Then, Q_t is an exponential moving average and $\lim_{t\to\infty} \mathbb{V}(Q_t) > 0$. Hence, Q_t cannot converge to q^*.

2. Exploring starts could help with condition 1. This also requires that all episodes end after finitely many steps. Otherwise, it could happen that condition 1 will still not be satisfied.

□

Question 3. (*Exercise 6.11 S&B*) Why is Q-learning considered an *off-policy control* method?

Solution. The book defines off-policy learning as a learning process when the value of a policy π is learned but the data is not generated from following π. Q-learning can learn the value function of an optimal policy π^* while following a policy that is different from π^*. As such, it is considered as an off-policy learning method.

Regarding whether Q-learning is a control method: In Chapter 5.3, we read that control methods are methods to approximate an optimal policy. Q-learning can be used for this purpose, and as such it is considered a control method.

□

Question 4. (*Exercise 6.12 S&B*, slightly modified) Suppose the action selection is greedy (as opposed to, say, ϵ-greedy).

1. Is Q-learning then exactly the same algorithm as Sarsa? That is, will they make exactly the same action selections and weight updates?

2. Are there any downsides to this choice?

Solution.

1. No, even in this case, Q-learning and Sarsa are different algorithms.

   Given the same starting state-action pair, if the action selection (during the episode) is greedy, Q-learning and Sarsa will make the same action selections and the same weight update (this is because, in case of greedy action selection, the next action $A' = \arg\max_a Q(S', a)$, and therefore $Q(S', A') = \max_a Q(S', a)$). However, this only happens on the first timestep. Recall that for the transition tuple (S, A, R, S', A'), Q-learning takes action A' after it makes the updates to the action values, whereas Sarsa takes the action A' before making the updates to the action values. This will result in different actions being chosen at later timesteps, and hence the data generated (and consequently the updates made) by the two algorithms would differ.

2. The downside is that there will be no exploration and as such neither Sarsa, nor Q-learning, can be expected to give good results.

□

Question 5. In this question we compare the variance of the target for Sarsa and Expected Sarsa. Recall that the update for Sarsa is

$$Q_{t+1}(S_t, A_t) = Q_t(S_t, A_t) + \alpha \left[R_{t+1} + \gamma Q_t(S_{t+1}, A_{t+1}) - Q_t(S_t, A_t) \right],$$

and the update for expected-Sarsa is

$$Q_{t+1}(S_t, A_t) = Q_t(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \sum_{a' \in \mathcal{A}} \pi(a'|S_{t+1}) Q_t(S_{t+1}, a') - Q_t(S_t, A_t) \right],$$

where π is a fixed policy that is used to generate the data $S_0, A_0, R_1, S_1, A_1, R_2, \ldots$.

1. Let $H_t = (S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_t)$ and $H'_t = (S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_t, A_t)$. Show that

$$\mathbb{V}[Q(S_{t+1}, A_{t+1}) \,|\, H_{t+1}] \geq \mathbb{V}\left[\sum_{a' \in \mathcal{A}} \pi(a'|S_{t+1}) Q(S_{t+1}, a') \,\middle|\, H_{t+1} \right].$$

2. **Challenge Question:** Show that

$$\mathbb{V}[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) \,|\, H'_t] \geq \mathbb{V}\left[R_{t+1} + \gamma \sum_{a' \in \mathcal{A}} \pi(a'|S_{t+1}) Q(S_{t+1}, a') \,\middle|\, H'_t \right],$$

that is, the appropriate conditional variance of the Sarsa target is always at least as large as that of for the Expected Sarsa target. (Hint: Use the law of total variance.)

Solution.

1. Clearly by Markov property, we can replace the variance conditioned on the history H_{t+1} by the variance conditioned on just the last state S_{t+1}. Then for any $s' \in \mathcal{S}$,

$$\mathbb{V}\left[\sum_{a' \in \mathcal{A}} \pi(a|S_{t+1}) Q(S_{t+1}, a') \,\middle|\, S_{t+1} = s' \right] = 0,$$

since there is no randomness in the expression. Then

$$\begin{aligned}
&\mathbb{V}\big[Q(S_{t+1}, A_{t+1}) \big| S_{t+1} = s'\big] \\
&= \mathbb{E}\big[Q(s', A_{t+1})^2 \big| S_{t+1} = s'\big] - \mathbb{E}\big[Q(s', A_{t+1}) \big| S_{t+1} = s'\big]^2 && \text{(definition of variance)} \\
&= \sum_{a' \in \mathcal{A}} \pi(a'|s') Q(s', a')^2 - \left(\sum_{a' \in \mathcal{A}} \pi(a'|s') Q(s', a') \right)^2 && \text{(by the definition of expectation and LOTUS)} \\
&\geq 0.
\end{aligned}$$

By Jensen's inequality, the equality in the above line holds if and only if the estimated Q-values at state s' are equal for all actions, that is, $Q(s', a_1) = \cdots = Q(s', a_{|\mathcal{A}|})$. (The above proof is essentially showing that the variance of a random variable is always non-negative.)

2. For convenience, we omit writing down the conditioning on $S_t = s$ and $A_t = a$ (that is, instead of writing $\mathbb{V}[\cdot|S_t = s, A_t = a]$, we write $\mathbb{V}[\cdot]$). We are asked to compare the variance of the target and

thus we compute:

$$\begin{aligned}
&\mathbb{V}\big[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})\big] - \mathbb{V}\Big[R_{t+1} + \gamma \sum_{a' \in \mathcal{A}} \pi(a'|S_{t+1}) Q(S_{t+1}, a')\Big] \\
&= \gamma^2 \Big\{ \mathbb{V}\big[Q(S_{t+1}, A_{t+1})\big] - \mathbb{V}\Big[\sum_{a' \in \mathcal{A}} \pi(a'|S_{t+1}) Q(S_{t+1}, a')\Big] \Big\} \\
&\qquad\qquad \text{(since } R_{t+1} \text{ is conditionally independent of } S_{t+1} \text{ and } A_{t+1}\text{)} \\
&= \gamma^2 \Big\{ \mathbb{E}\Big[\mathbb{V}\big[Q(S_{t+1}, A_{t+1})\big|S_{t+1}\big]\Big] - \mathbb{V}\Big[\mathbb{E}\big[Q(S_{t+1}, A_{t+1})\big|S_{t+1}\big]\Big] \\
&\qquad - \mathbb{E}\Big[\mathbb{V}\Big[\sum_{a' \in \mathcal{A}} \pi(a'|S_{t+1}) Q(S_{t+1}, a')\Big|S_{t+1}\Big]\Big] + \mathbb{V}\Big[\mathbb{E}\Big[\sum_{a' \in \mathcal{A}} \pi(a'|S_{t+1}) Q(S_{t+1}, a')\Big]\Big|S_{t+1}\Big]\Big\}. \\
&\qquad\qquad \text{(by the law of total variance)}
\end{aligned}$$

Now, notice that $\mathbb{E}\Big[\mathbb{V}\big[\sum_{a' \in \mathcal{A}} \pi(a|S_{t+1}) Q(S_{t+1}, a')\big|S_{t+1} = s'\big]\Big] = 0$. Also,

$$\begin{aligned}
\mathbb{V}\Big[\mathbb{E}\big[Q(S_{t+1}, A_{t+1})\big|S_{t+1}\big]\Big] &= \mathbb{V}\Big[\sum_{a' \in \mathcal{A}} \pi(a'|S_{t+1}) Q(S_{t+1}, a')\Big] \\
&= \mathbb{V}\Big[\mathbb{E}\Big[\sum_{a' \in \mathcal{A}} \pi(a'|S_{t+1}) Q(S_{t+1}, a')\Big|S_{t+1}\Big]\Big].
\end{aligned}$$

Putting the above results into the previous equation (and now explicitly writing the conditioning on $S_t = s$ and $A_t = a$) gives us

$$\begin{aligned}
&\mathbb{V}\Big[Q^{\text{SARSA}}(S_t, A_t)\Big|S_t = s, A_t = a\Big] - \mathbb{V}\Big[Q^{\text{expected-SARSA}}(S_t, A_t)\Big|S_t = s, A_t = a\Big] \\
&= \gamma^2 \mathbb{E}\Big[\mathbb{V}\big[Q(S_{t+1}, A_{t+1})\big|S_{t+1}, S_t = s, A_t = a\big]\Big|S_t = s, A_t = a\Big] \\
&= \gamma^2 \mathbb{E}\Big[\mathbb{V}\big[Q(S_{t+1}, A_{t+1})\big|S_{t+1}\big]\Big|S_t = s, A_t = a\Big] \qquad \text{(using the Markov property)} \\
&\geq 0. \qquad \text{(since the variance is always positive)}
\end{aligned}$$

Therefore, we conclude that the SARSA update rule has a higher variance than the expected-SARSA update rule when following the same target policy.

□

Question 6. (Challenge Question) (*Exercise 6.13 S&B*) What are the update equations for Double Expected Sarsa with an ϵ-greedy target policy?

Solution. Double Expected Sarsa will maintain two action-value functions, $Q_t^{(i)}$, $i = 1, 2$. Given any function $q : \mathcal{S} \times \mathcal{A} \to R$, let π_q denote the the ϵ-greedy policy with respect to q: $\pi_q(a|s) = (1-\epsilon)\mathbb{I}(a = \arg\max_{a'} q(s,a)) + \epsilon/A$, where $A = |\mathcal{A}|$ is the number of actions. Here, for simplicity, we define arg max so that it breaks ties in a systematic fashion. For arbitrary $q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and memoryless policy π, introduce the notation $q(s,\pi) := \sum_a \pi(a|s)q(s,a)$. The update rule then is as follows: Choose I_t uniformly at random from $\{1, 2\}$. Then, one possible update rule is

$$Q_{t+1}^{(I_t)}(S_t, A_t) = (1-\alpha)Q_t^{(I_t)}(S_t, A_t) + \alpha\left[R_{t+1} + \gamma Q_t^{(I_t)}(S_{t+1}, \pi_{Q_t^{(3-I_t)}})\right].$$

Another possible update rule is

$$Q_{t+1}^{(I_t)}(S_t, A_t) = (1-\alpha)Q_t^{(I_t)}(S_t, A_t) + \alpha\left[R_{t+1} + \gamma Q_t^{(3-I_t)}(S_{t+1}, \pi_{Q_t^{(I_t)}})\right].$$

Note that in both update rules, following the idea of double-Q learning, the policy used in the calculation of the "target" uses the action-value estimate ($Q_t^{(3-I_t)}$ in the first update rule, $Q_t^{(I_t)}$ in the second update rule) that differs from the action-value estimate ($Q_t^{(I_t)}$ in the first update rule, $Q_t^{(3-I_t)}$ in the second update rule) used to get the target itself.

□

Question 7. (*Exercise 6.8 S&B*) Show that an action-value version of the expression

$$G_t - V(S_t) = \sum_{k=t}^{T-1} \gamma^{k-t} \delta_k^{(\text{value})},$$

where $\delta^{(\text{value})} = R_{t+1} + \gamma V(S_{t+1}, A_{t+1}) - V(S_t, A_t)$, holds for the action-value form of the TD error

$$\delta_t = R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)$$

as well. (Assume that the action-values don't change from step to step, that is $Q_t = Q$ for all $t \geq 0$ and some function Q.)

Solution. Let the terminal timestep be denoted by T. Therefore, S_T represents the terminal state, and $G_T = Q(S_T, a) = 0$ for all actions $a \in \mathcal{A}$. Then

$$\begin{aligned}
G_t - Q(S_t, A_t) &= R_{t+1} + \gamma G_{t+1} - Q(S_t, A_t) + \gamma Q(S_{t+1}, A_{t+1}) - \gamma Q(S_{t+1}, A_{t+1}) \\
&= \Big(R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\Big) + \gamma\Big(G_{t+1} - Q(S_{t+1}, A_{t+1})\Big) \\
&= \delta_t + \gamma\Big(G_{t+1} - Q(S_{t+1}, A_{t+1})\Big) \\
&= \delta_t + \gamma\delta_{t+1} + \gamma^2\Big(G_{t+2} - Q(S_{t+2}, A_{t+2})\Big) \\
&\vdots \\
&= \delta_t + \gamma\delta_{t+1} + \gamma^2\delta_{t+2} + \cdots + \gamma^{T-t-1}\delta_{T-1} + \gamma^{T-t}\Big(G_T - Q(S_T, \cdot)\Big) \\
&= \sum_{k=t}^{T-1} \gamma^{k-t}\delta_k.
\end{aligned}$$

□

Question 8. Solve the following questions.

1. Let $X_1, X_2, \ldots$ be independent random variables, that take on the values $\{0, 1\}$. Assume that for all $t \geq 1$, $\mathbb{P}(X_t = 1) = p$, where $p \in (0, 1)$. Define

$$T := \min\{t \geq 1 \,:\, X_t = 1\}\,.$$

   Think of $X_1, X_2, \ldots$ as outcomes of a coin-flip, where 0 corresponds to tails and 1 corresponds to heads. Then T is the number of flips until seeing the first head (including the first timestep). Show that

$$\mathbb{E}[T] = 1/p.$$

2. Using the answer in the first part, show that there exists a finite, episodic MDP with n states and two actions, such that all of the following hold:

   - The episode lengths are *at least* n.
   - New episodes start from a fixed state s_0 of the MDP.
   - ϵ-greedy with Q-learning (initialized at zero) needs at least $\Omega(2^n)$ episodes before the action-value of the optimal action at s_0 gets higher than the action value of the sub-optimal action at s_0, regardless of the choice of the stepsizes in Q-learning.
   - The value of s_0 under the optimal policy is one.

Solution.

1. Fix a number $k \geq 1$. Then,

$$\begin{aligned}
\mathbb{P}(T = k) &= \mathbb{P}(X_1 = 0, X_2 = 0, \ldots, X_{k-1} = 0, X_k = 1) \\
&= \left(\prod_{i=1}^{k-1} \mathbb{P}(X_i = 0)\right) \cdot \mathbb{P}(X_k = 1) \qquad \text{(since } X_i\text{s are independently distributed)} \\
&= (1-p)^{k-1} p.
\end{aligned}$$

   Therefore,

$$\mathbb{E}[T] = \sum_{k=1}^{\infty} k \cdot \mathbb{P}(T = k) = \sum_{k=1}^{\infty} k(1-p)^{k-1} p = p \sum_{k=1}^{\infty} k(1-p)^{k-1}.$$

   Note that $(1-p)\mathbb{E}[T] = p\sum_{k=1}^{\infty} k(1-p)^k = p\sum_{k=1}^{\infty}(k-1)(1-p)^{k-1}$. Then,

$$\mathbb{E}[T] - (1-p)\mathbb{E}[T] = p\mathbb{E}[T] = p\sum_{k=1}^{\infty}(1-p)^{k-1} = \frac{p}{1-(1-p)} = 1 \qquad \Rightarrow \qquad \mathbb{E}[T] = 1/p.$$

2. Let $\mathcal{S} = \{0, 1, \ldots, n-1, n\}$, $\mathcal{A} = \{0, 1\}$, n is the terminal state. The transitions are deterministic; the next state is $\min(n, s + a)$ when the current state is $s \in \mathcal{S}$ and the action is $a \in \mathcal{A}$. All rewards are zero, except when taking action 1 at state $n - 1$. The initial state is $s_0 = 0$.

   Clearly, the episode length are at most n. It remains to see the behavior of Q-learning with ϵ-greedy when the initial value estimates are zero. Due to the Q-learning update, all action values remain zero until state n is visited for the first time (because all rewards incurred are zero unless one transitions to state n). Let T be the index of the first episode when state n is encountered for the first time. If T'

is the index of the first episode when the action value of 1 at state 0 is higher than the action value of action 0 at state 0, we have $T' > T$ because in episode T this property does not hold.

When all action values are the same, Q-learning with ϵ-greedy chooses an action uniformly at random from the two actions. Thus, in every episode before episode T, Q-learning chooses actions uniformly at random. While following the uniform random policy, the probability of encountering state n is $p = (1/2)^n$, because for this to happen, action 1 has to be chosen n times and the probability of choosing action 1 is $1/2$, and the choices are independent of each other. Thus, $\mathbb{E}[T] = 1/p = 2^n$ by the first part of the problem, and hence from $T' > T$, we have

$$\mathbb{E}[T'] > \mathbb{E}[T] = 2^n \,,$$

which means that $\mathbb{E}[T'] = \Omega(2^n)$.

□
