Shaurya
2016194

RL Assignment-3

Ans-1 (Ex 5.4)

$$G_n = \frac{R_1 + R_2 \dots + R_n}{n}$$

$$= \frac{(n-1)G_{n-1} + R_n}{n}$$

$$\Rightarrow G_n = G_{n-1} + \frac{1}{n}(R_n - G_{n-1})$$

we can maintain an array for n, which stores count of each (S_t, A_t)

Pseudocode can be altered by —

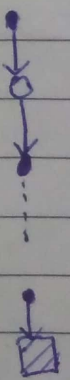
⋮

unless the pair S_t, A_t appears in $S_0, A_0, S_1 \dots S_{t-1}, A_{t-1}$:

$n(S_t, A_t) \leftarrow n(S_t, A_t) + 1$

$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{1}{n(S_t, A_t)} \left[R_{t+1} - Q(S_t, A_t) \right]$

Ans-2 (Ex 5.3)



Ans-3 (Ex 5.6)

Let $T(t)$ denote termination time after t.

$\tau(s,a)$ denote all time steps of first visit to (s,a) within an episode

G_t denote return after t upto $T(t)$

Then, $Q(s,a) = \dfrac{\sum_{t \in \tau(s,a)} \rho_{t:T-1} G_t}{\sum_{t \in \tau(s,a)} \rho_{t:T-1}}$

where $\rho_{t:T-1} = \prod_{k=t}^{T-1} \dfrac{\pi(A_k|S_k)}{b(A_k|S_k)}$

Ans-8 (Exc 6.12) As action selection is greedy, both SARSA & Q-learning become same as they both use same form of ϵ-soft policy for exploration.
update eq. of both become same -

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left(R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\right)$$

A_{t+1} selected greedily.

They will choose same actions & have same weight updates.

Ans-6 (Ex-6.3) In the first episode, the agent ended its episode by moving to the extreme left terminal state.

$$V(S_t) \leftarrow V(S_t) + \alpha\left[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right]$$

$\alpha = 0.1$ & $\gamma = 1$ & $V(s) = 0.5$ $\forall s$

$$\Rightarrow V(S_t) \leftarrow V(S_t) + 0.1\left[R_{t+1} + (1)V(S_{t+1}) - V(S_t)\right]$$

$$\Rightarrow V(S_t) \leftarrow V(S_t) + 0.1\left[R_{t+1} + V(S_{t+1}) - V(S_t)\right]$$

As initial $V(s) = 0.5$ for all states except terminal state where V is 0 & reward is 0 for all transitions.

$\Rightarrow V(S_t) \leftarrow V(S_t)$ when S_{t+1} is not terminal

Hence V remains same except when S_{t+1} is terminal.

If S_{t+1} is terminal $\Rightarrow S_t = a$

$$V(a) \leftarrow V(a) + 0.1[0 + 0 - 0.5]$$

$$\Rightarrow V(a) \leftarrow V(a) - 0.05 = 0.45$$

Hence V(a) changes by -0.05.

Ans Ex-6.4 Yes, conclusion about which algorithm is better would have changed if ~~different~~ wide range of α were used. This is because different alpha correspond to different convergent rates. If we keep α very small, then, both algorithms would perform significantly better as taking small steps we avoid overshooting the maxima, but our convergence would have been slower.

Ex-6.5 The error first goes down then goes up again at high alphas in TD because high α's overshoots the maxima & keeps oscillating around the maxima.
This also depends on the initialisation of value function as even with small α, the difference $\left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right)$ could be large if value function was initialised arbitrarily. This large difference could help to overshoot the maxima if not taken care.

Ans-5 TD updates are better on average compared to Monte Carlo eg-
Suppose there is a new ~~bad~~ terminal state which gives high +ve reward. As TD uses step reward to estimate V, value function for all states can be updated rather quickly. In MC, we use return to update the value function. States close to the ~~terminal states~~ start states will take more iterations to update due to noisy nature of the return.