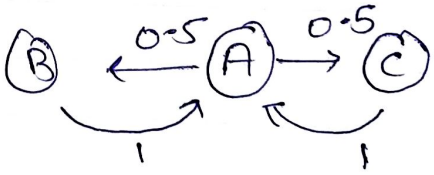
Q1.

Since, we have no idea about underlying Transition probs. I would use Algorithm Y. with Monte-Carlo control Method.
If we were to use Algo X. we would need to know all the possible transitions to all states, to find out optimal policy. But Monte-Carlo Methods don't need to know any probability transitions. and work with action values.

Q.3

Stationary-Policy: refers to a policy which is unchanging through time. i.e., the probabilities of selecting the actions doesn't change through time, But the actions selected might be different, based on probabilities.

Ex: when in state 'A'.
the policy is go to B or C with prob 0.5.



we can go to state A many times but the action selected might be different each time.

Date

Expt. No. Page No.

Q4.

$$A_t = \arg\max_a \left(Q_t(a) + c\sqrt{\frac{\ln t}{n_t(a)}} \right)$$

Let $c = 2$

$t = 25$

We know:

Arm 1: $n_1 = 8$, $Q_{25}(1) = 0.41$

$$\Rightarrow 0.41 + 2\sqrt{\frac{\log 25}{8}} = 1.678$$

Arm 2: $n_2 = 4$, $Q_{25}(n_2) = 0.71$

$$= 2.50$$

Arm 3: $n_3 = 7$, $Q_{25}(n_3) = 0.64$

$$= 1.996$$

Arm 4: $n_4 = 3$, $Q_{25}(n_4) = 0.75$

$$= 2.821$$

Arm 5: $n_5 = 3$, $Q_{25}(n_5) = 0.80$

$$= 2.871$$

∴ The Arm selected is Arm 5

Teacher's Signature :

Q6. Generalized Policy Iteration is a method of method of evaluating & improving the ~~policy~~ policy recurrently. First we evaluate our policy, then based on it we improve our policy and then we ~~evaluate~~ evaluate our policy & so on. As the number of ~~time~~ iterations progress, both ~~value~~ state value & policy converges to an optimal value & optimal policy.

Q8. Monte-Carlo Exploring Starts, has the trivial problem of exploring all starts. If the policy is deterministic, then ~~the~~ our ES method cannot capture all different policy variations. So, our ~~epsi~~ episode generation policy has to be stochastic. The other problem is it has to have infinite episodes to converge to an optimal policy. (not really practical).

Date

Expt. No. Page No.

Q 10.

The computation could be done using Dynamic programming Method. But the easier way is to check the rewards.

If we start at C, with 0.5 prob. we go left or right, but if we go right, we get reward 1 at end.

So. $V(C) = E(\text{reward}) = 0.5 \times 1 = 0.5$

Similarly, from E, we can see that

$$V(E) = V(E) \times P(\text{right}) + \frac{1}{2} \times V(D)$$

$$= \frac{1}{2} + \frac{1}{4}[V(C) + V(E)].$$

So, $V(E) = \frac{5}{6}$, Then we get $V(D) = \frac{4}{6}$.

Similarly we can calculate for A, B $= \frac{1}{6}, \frac{2}{6}$.

Teacher's Signature :

Q.13

In Both SARSA & Q-learning are model less learning methods.

In sarsa, the action a is taken following the ϵ-greedy policy. But in Q-learning, we choose the action 'a' which maximizes the Q-value at the new state $\in Q(s', a)$.

update for SARSA:

$$Q(s,a) = Q(s,a) + \alpha(R + \gamma Q(s',a') - Q(s,a)$$

update for Q-learning:

$$Q(s,a) = Q(s,a) + \alpha(R + \gamma \max_{a'} Q(s',a') - Q(s,a)$$

→ SARSA will learn the optimal ~~Policy~~ ϵ-greedy policy. We expect that if $\epsilon \to 0$, the SARSA will eventually learn the optimal policy, But Q-learning always learns ~~the~~ or converge to the optimal q_*.

Date

Expt. No.

Page No.

Q11.

In Q-learning we select the next action greedily, like select the action which maximizes

Q11.

In Q-learning we update the Q-values greedily, like choose a greedy action a', which maximizes the Q-value next state.

∴ The update rule is:

$$Q(s,a) = Q(s,a) + \alpha\left[R + \gamma \max_{a'} Q(s,a') - Q(s,a)\right]$$

$$s \leftarrow s'$$

To convert it to SARSA, we remove the greedy selection of Q-value and do an on-policy iteration, where select the next action by using ϵ-greedy.

∴ The update rule changes to.

$$Q(s,a) = Q(s,a) + \alpha\left(R + \gamma Q(s',A') - Q(s,A)\right]$$

$$s \leftarrow s', \quad A \leftarrow A'$$

Teacher's Signature :

Q.7.

The three MC, TD & DP learns from experiences, They all use the concept of value iteration, policy iteration or Generalized policy iteration to converge to an optimal policy. Though, DP requires, transitional probs to be given, MC & TD doesn't need them.

Q.5

Since, 14 is the ~~last~~ state before terminal state. 14, right takes us to terminal state. The action value

$q_\pi(14, \text{right}) = -1.$