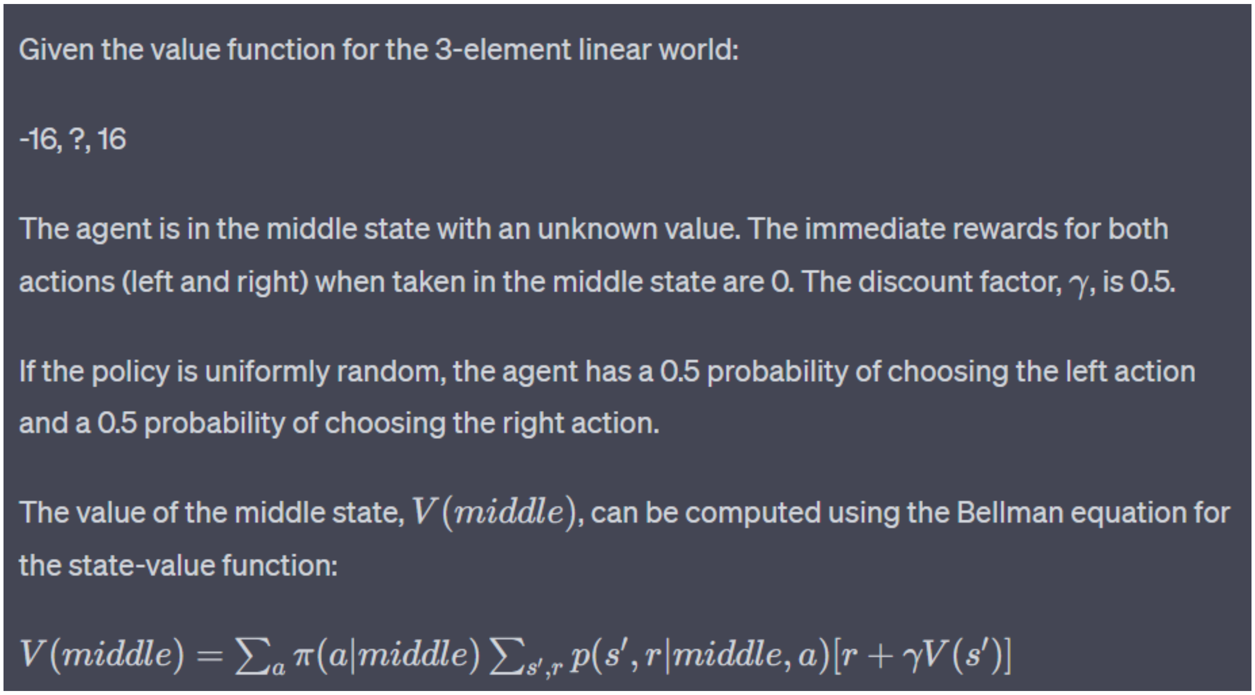
Question 1:



Given the value function for the 3-element linear world:

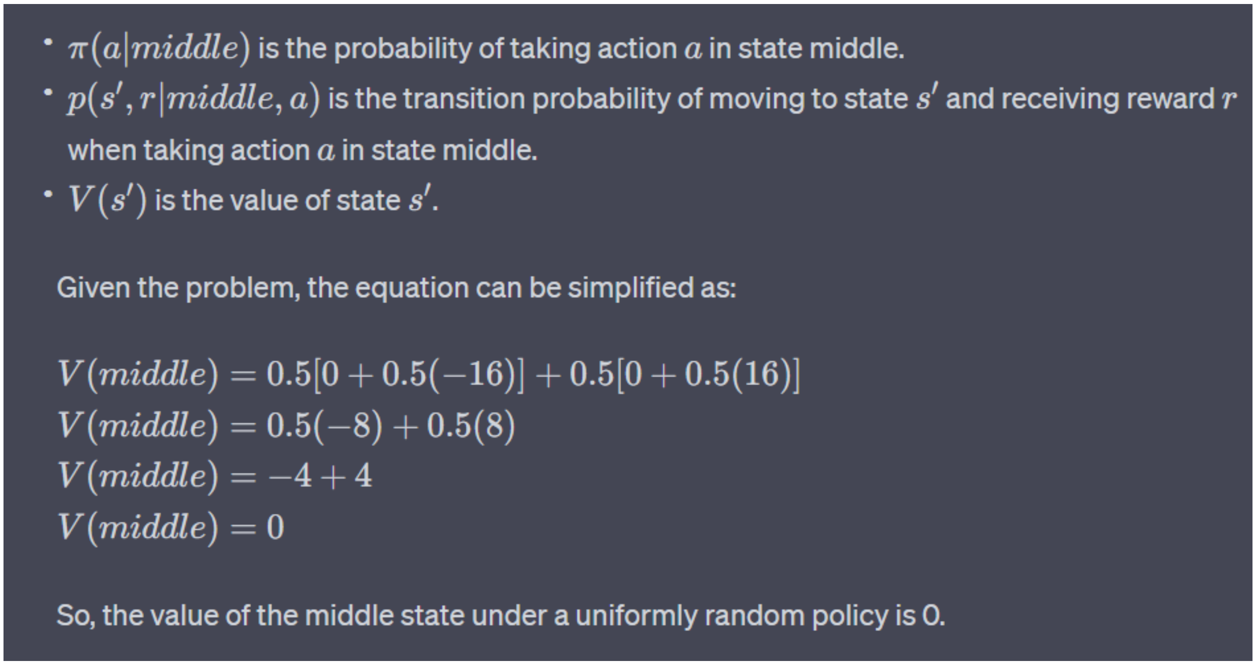
-16, ?, 16

The agent is in the middle state with an unknown value. The immediate rewards for both actions (left and right) when taken in the middle state are 0. The discount factor, γ, is 0.5.

If the policy is uniformly random, the agent has a 0.5 probability of choosing the left action and a 0.5 probability of choosing the right action.

The value of the middle state, $V(middle)$, can be computed using the Bellman equation for the state-value function:

$$V(middle) = \sum_a \pi(a|middle) \sum_{s',r} p(s', r|middle, a)[r + \gamma V(s')]$$



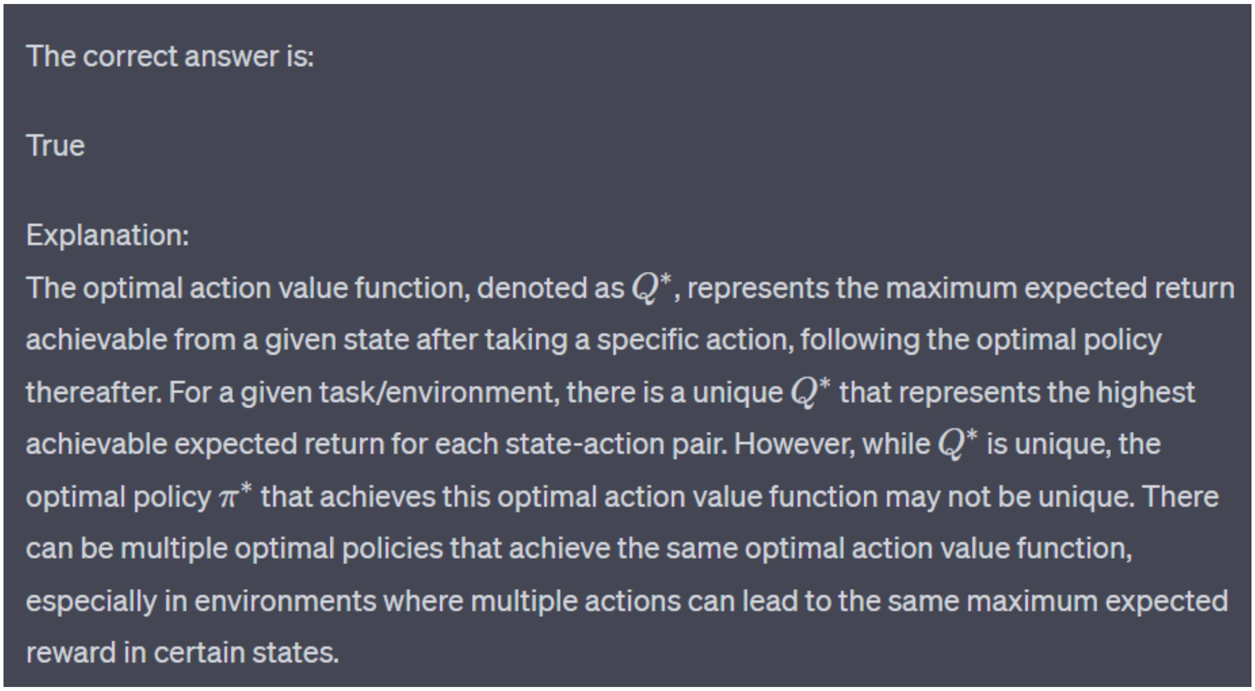
- $\pi(a|middle)$ is the probability of taking action a in state middle.
- $p(s', r|middle, a)$ is the transition probability of moving to state s' and receiving reward r when taking action a in state middle.
- $V(s')$ is the value of state s'.

Given the problem, the equation can be simplified as:

$$V(middle) = 0.5[0 + 0.5(-16)] + 0.5[0 + 0.5(16)]$$
$$V(middle) = 0.5(-8) + 0.5(8)$$
$$V(middle) = -4 + 4$$
$$V(middle) = 0$$

So, the value of the middle state under a uniformly random policy is 0.

Question 2:

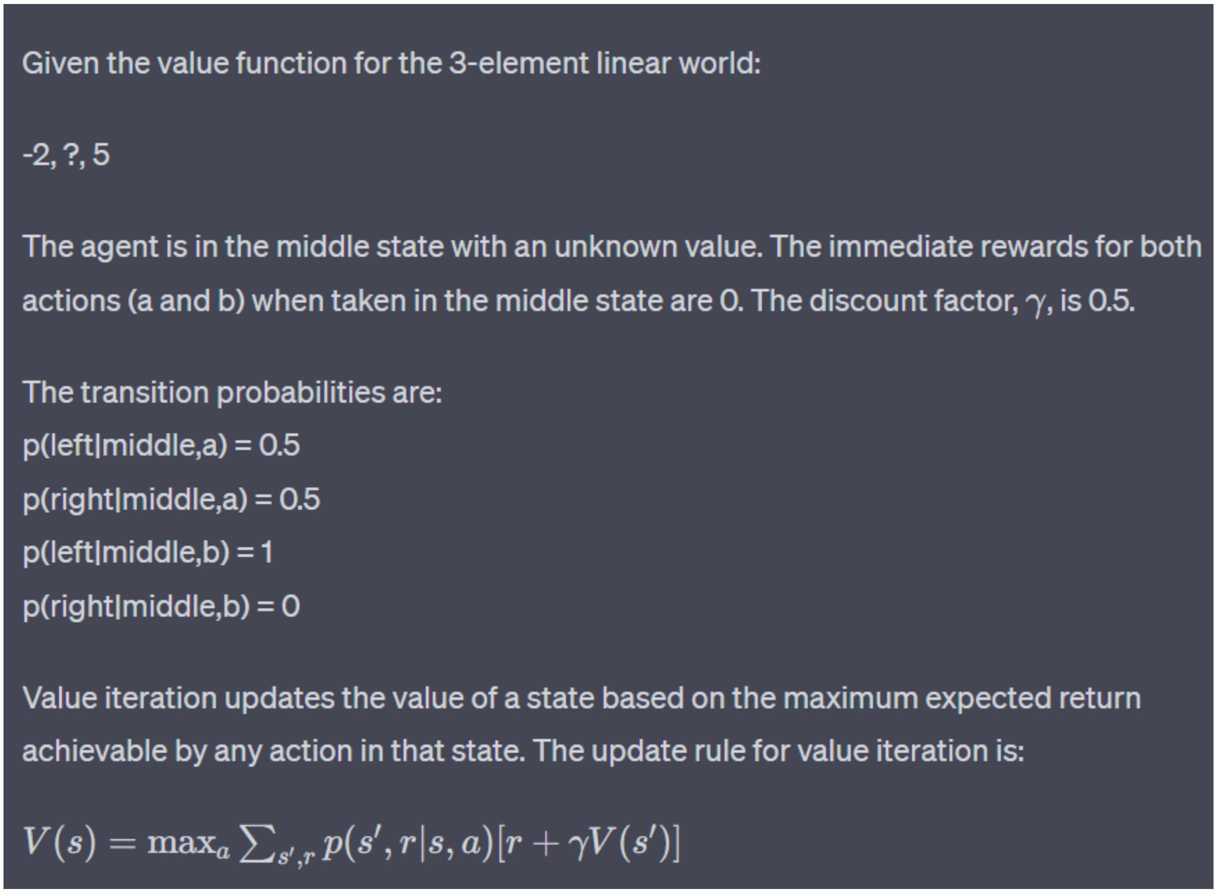


The correct answer is:

True

Explanation:
The optimal action value function, denoted as Q^*, represents the maximum expected return achievable from a given state after taking a specific action, following the optimal policy thereafter. For a given task/environment, there is a unique Q^* that represents the highest achievable expected return for each state-action pair. However, while Q^* is unique, the optimal policy π^* that achieves this optimal action value function may not be unique. There can be multiple optimal policies that achieve the same optimal action value function, especially in environments where multiple actions can lead to the same maximum expected reward in certain states.

Question 3:



Given the value function for the 3-element linear world:

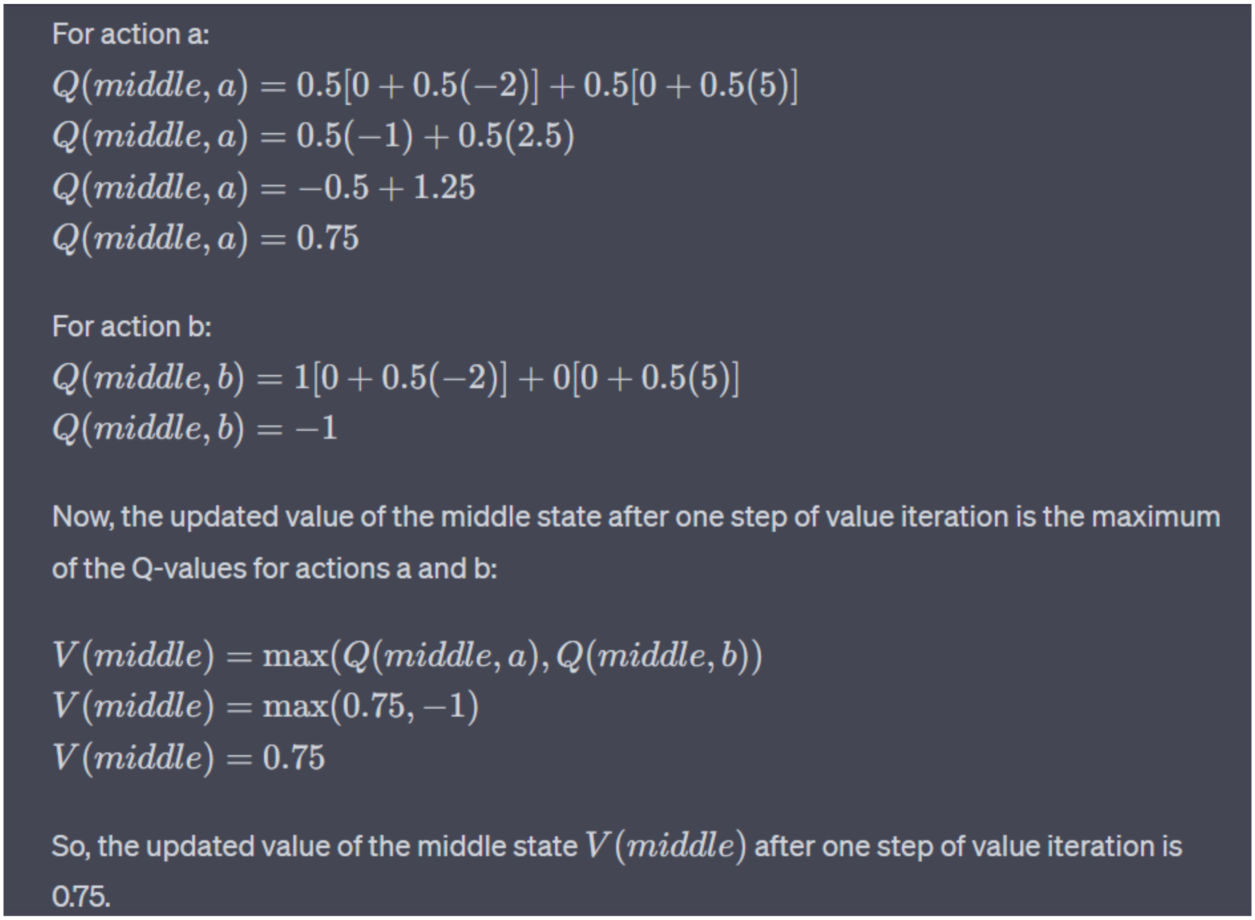
-2, ?, 5

The agent is in the middle state with an unknown value. The immediate rewards for both actions (a and b) when taken in the middle state are 0. The discount factor, γ, is 0.5.

The transition probabilities are:
p(left|middle,a) = 0.5
p(right|middle,a) = 0.5
p(left|middle,b) = 1
p(right|middle,b) = 0

Value iteration updates the value of a state based on the maximum expected return achievable by any action in that state. The update rule for value iteration is:

$$V(s) = \max_a \sum_{s',r} p(s', r|s, a)[r + \gamma V(s')]$$



For action a:

$$Q(middle, a) = 0.5[0 + 0.5(-2)] + 0.5[0 + 0.5(5)]$$
$$Q(middle, a) = 0.5(-1) + 0.5(2.5)$$
$$Q(middle, a) = -0.5 + 1.25$$
$$Q(middle, a) = 0.75$$

For action b:

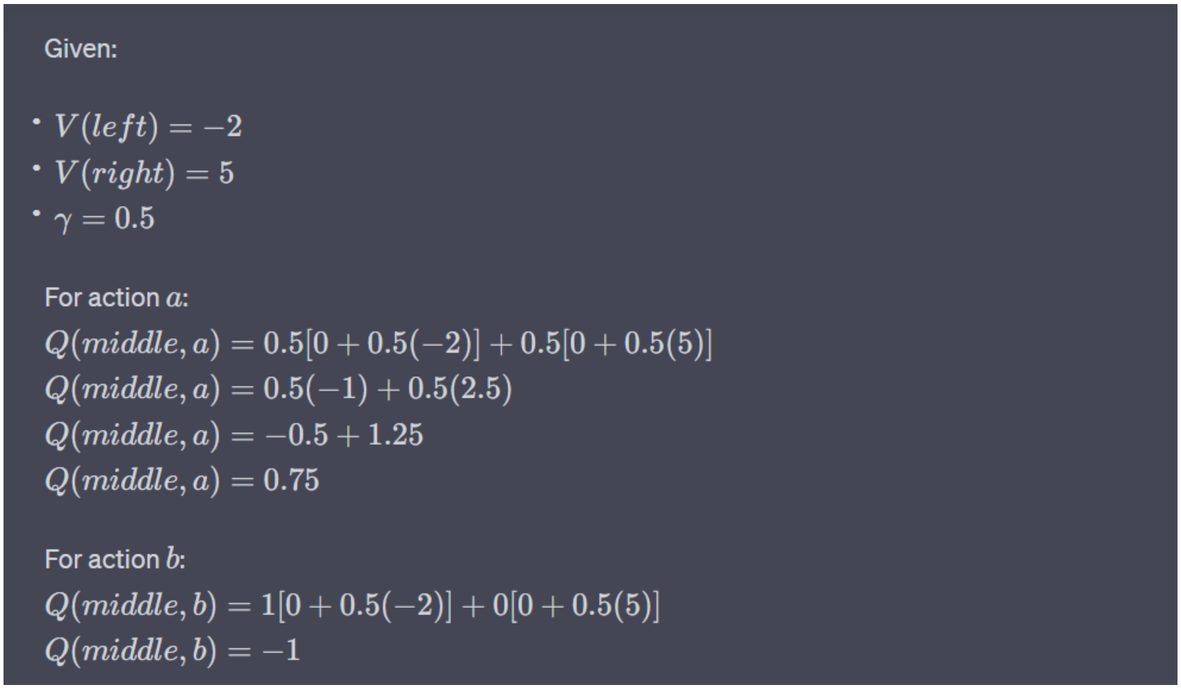
$$Q(middle, b) = 1[0 + 0.5(-2)] + 0[0 + 0.5(5)]$$
$$Q(middle, b) = -1$$

Now, the updated value of the middle state after one step of value iteration is the maximum of the Q-values for actions a and b:

$$V(middle) = \max(Q(middle, a), Q(middle, b))$$
$$V(middle) = \max(0.75, -1)$$
$$V(middle) = 0.75$$

So, the updated value of the middle state $V(middle)$ after one step of value iteration is 0.75.

Question 4:



Given:

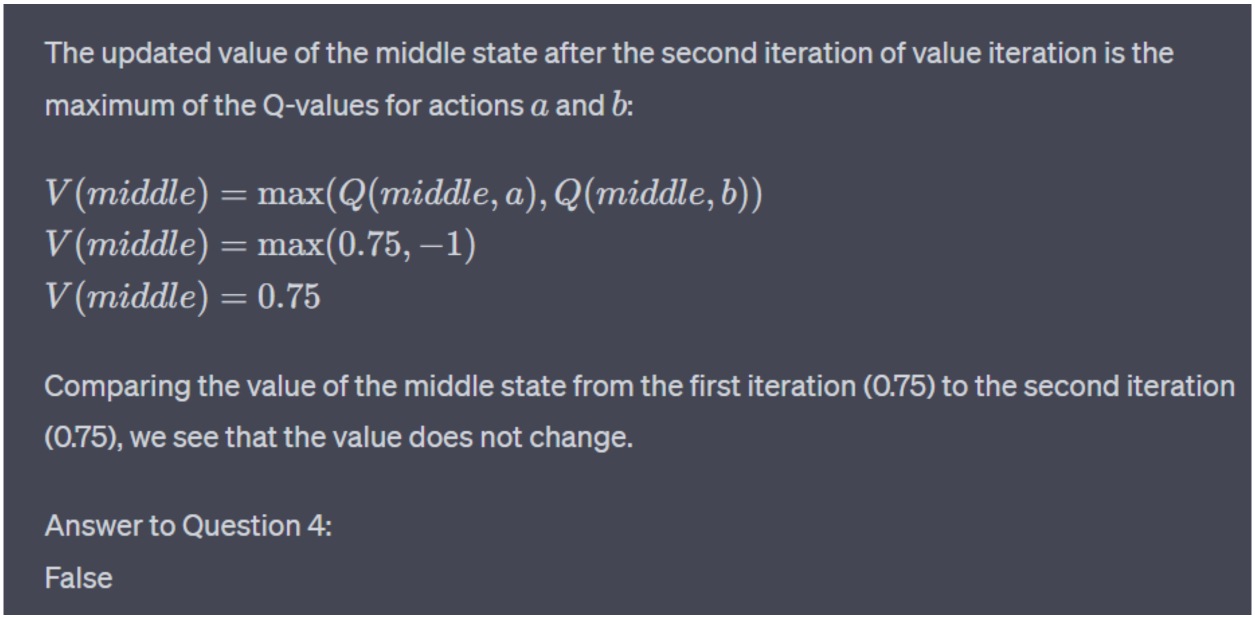
- $V(left) = -2$
- $V(right) = 5$
- $\gamma = 0.5$

For action a:

$$Q(middle, a) = 0.5[0 + 0.5(-2)] + 0.5[0 + 0.5(5)]$$
$$Q(middle, a) = 0.5(-1) + 0.5(2.5)$$
$$Q(middle, a) = -0.5 + 1.25$$
$$Q(middle, a) = 0.75$$

For action b:

$$Q(middle, b) = 1[0 + 0.5(-2)] + 0[0 + 0.5(5)]$$
$$Q(middle, b) = -1$$



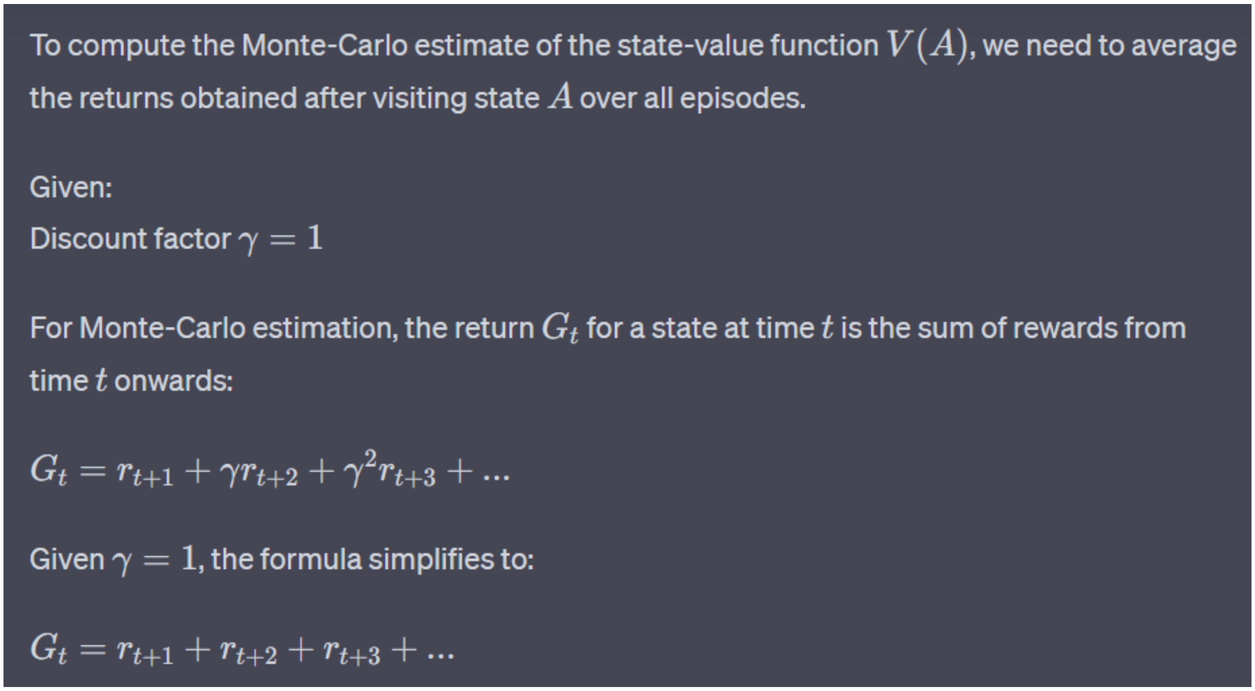
The updated value of the middle state after the second iteration of value iteration is the maximum of the Q-values for actions a and b:

$$V(middle) = \max(Q(middle, a), Q(middle, b))$$
$$V(middle) = \max(0.75, -1)$$
$$V(middle) = 0.75$$

Comparing the value of the middle state from the first iteration (0.75) to the second iteration (0.75), we see that the value does not change.

Answer to Question 4:
False

Question 5:



To compute the Monte-Carlo estimate of the state-value function $V(A)$, we need to average the returns obtained after visiting state A over all episodes.

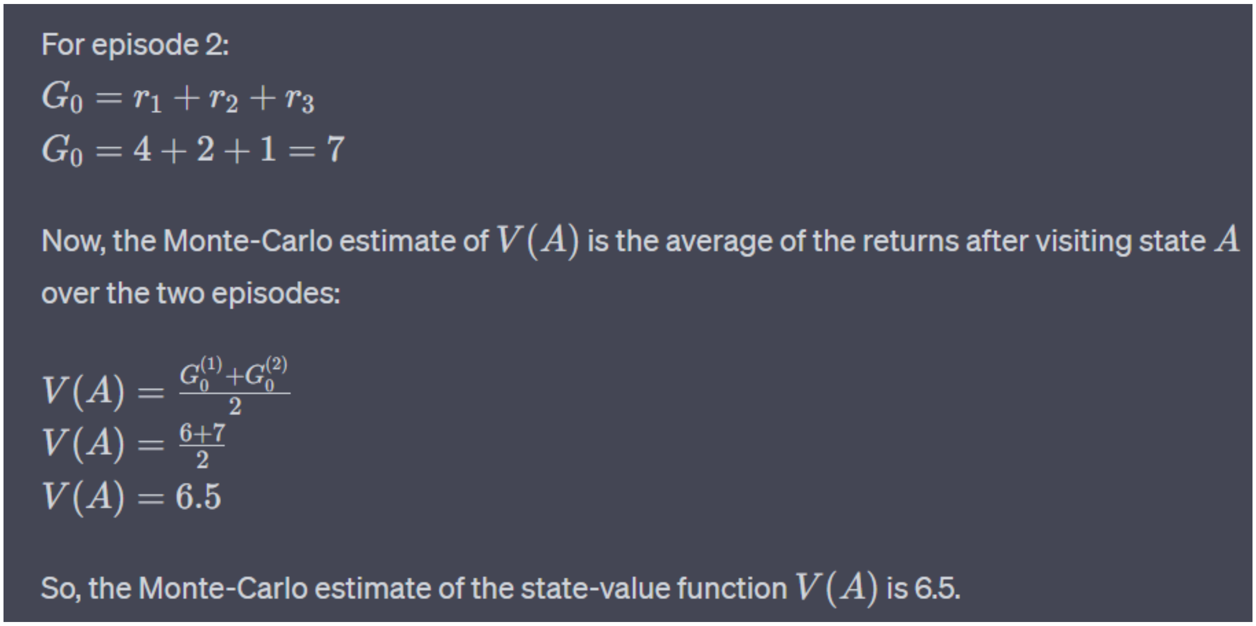
Given:
Discount factor $\gamma = 1$

For Monte-Carlo estimation, the return G_t for a state at time t is the sum of rewards from time t onwards:

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ...$$

Given $\gamma = 1$, the formula simplifies to:

$$G_t = r_{t+1} + r_{t+2} + r_{t+3} + ...$$



For episode 2:

$G_0 = r_1 + r_2 + r_3$

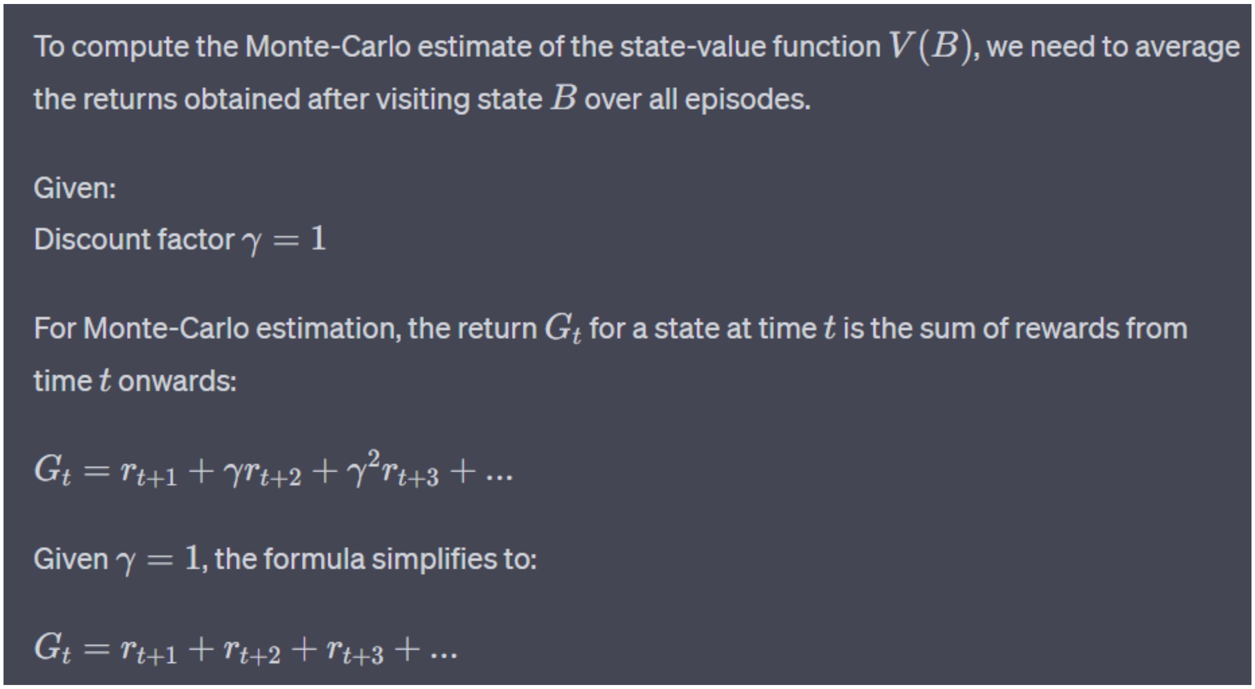
$G_0 = 4 + 2 + 1 = 7$

Now, the Monte-Carlo estimate of $V(A)$ is the average of the returns after visiting state A over the two episodes:

$$V(A) = \frac{G_0^{(1)} + G_0^{(2)}}{2}$$
$$V(A) = \frac{6+7}{2}$$
$$V(A) = 6.5$$

So, the Monte-Carlo estimate of the state-value function $V(A)$ is 6.5.

Question 6:



To compute the Monte-Carlo estimate of the state-value function $V(B)$, we need to average the returns obtained after visiting state B over all episodes.

Given:

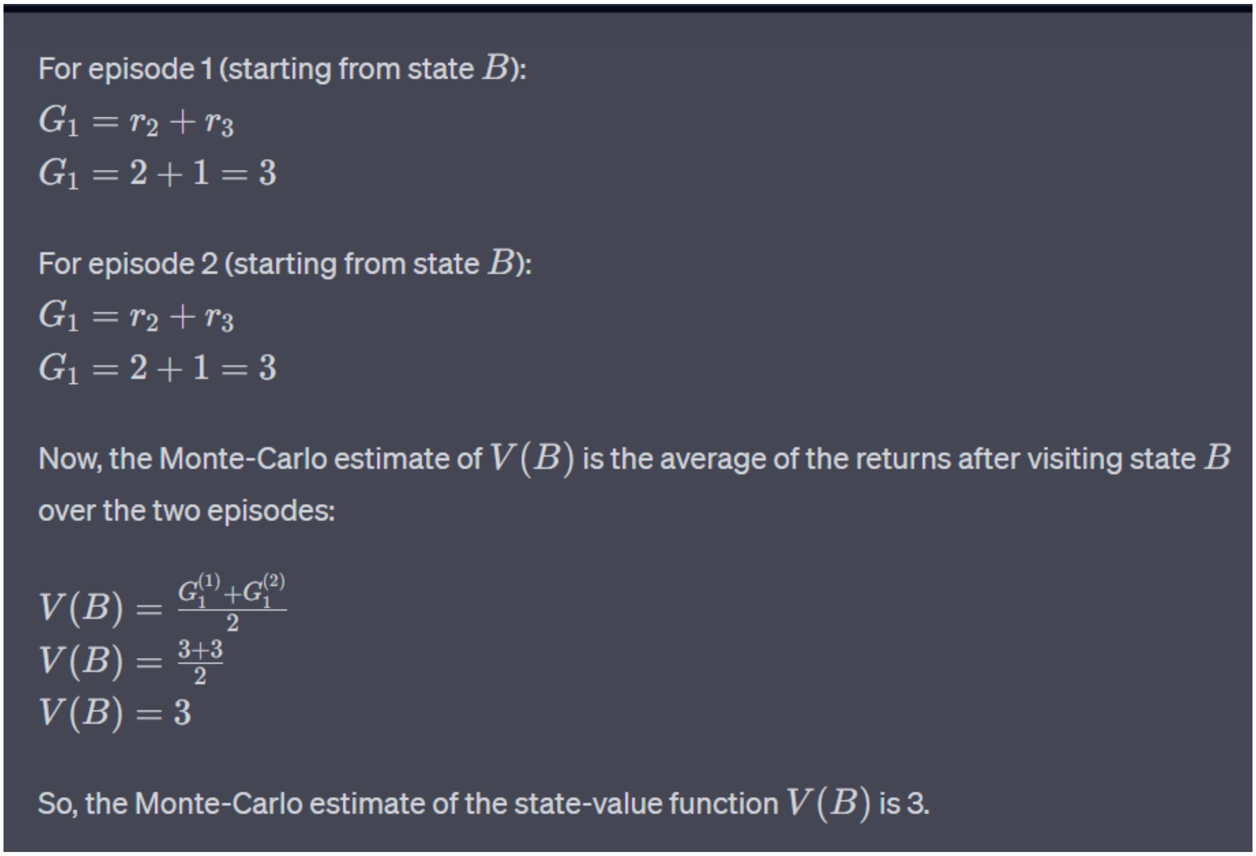
Discount factor $\gamma = 1$

For Monte-Carlo estimation, the return G_t for a state at time t is the sum of rewards from time t onwards:

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ...$$

Given $\gamma = 1$, the formula simplifies to:

$$G_t = r_{t+1} + r_{t+2} + r_{t+3} + ...$$



For episode 1 (starting from state B):

$G_1 = r_2 + r_3$

$G_1 = 2 + 1 = 3$

For episode 2 (starting from state B):

$G_1 = r_2 + r_3$

$G_1 = 2 + 1 = 3$

Now, the Monte-Carlo estimate of $V(B)$ is the average of the returns after visiting state B over the two episodes:

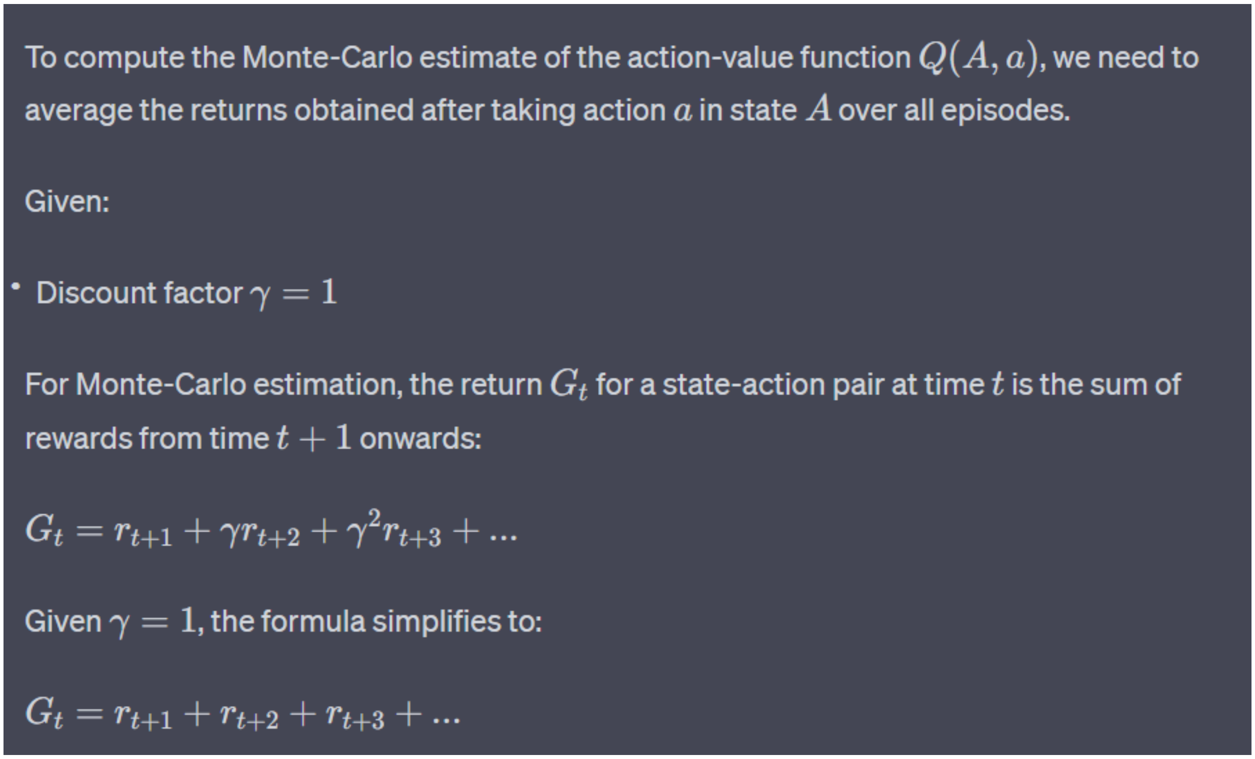
$V(B) = \frac{G_1^{(1)} + G_1^{(2)}}{2}$

$V(B) = \frac{3+3}{2}$

$V(B) = 3$

So, the Monte-Carlo estimate of the state-value function $V(B)$ is 3.

Question 7:



To compute the Monte-Carlo estimate of the action-value function $Q(A, a)$, we need to average the returns obtained after taking action a in state A over all episodes.

Given:

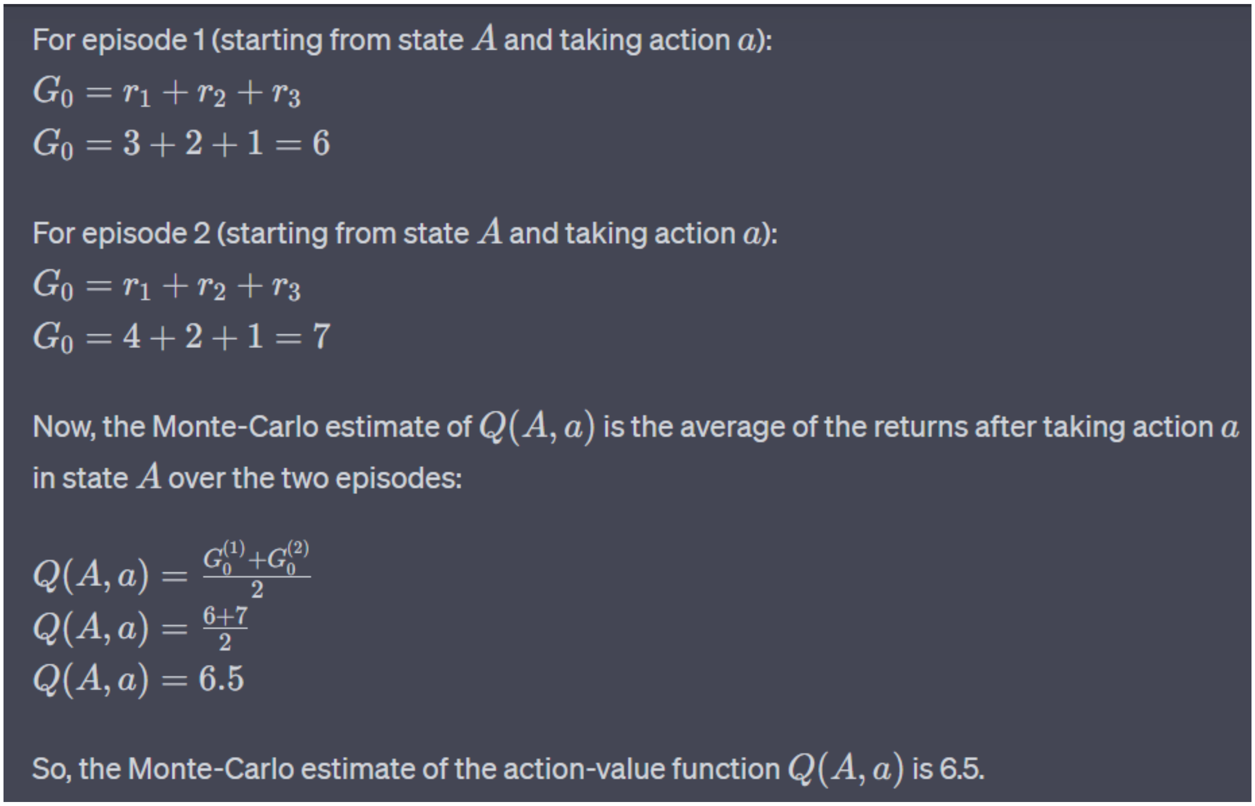
- Discount factor $\gamma = 1$

For Monte-Carlo estimation, the return G_t for a state-action pair at time t is the sum of rewards from time $t + 1$ onwards:

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ...$$

Given $\gamma = 1$, the formula simplifies to:

$$G_t = r_{t+1} + r_{t+2} + r_{t+3} + ...$$



For episode 1 (starting from state A and taking action a):

$G_0 = r_1 + r_2 + r_3$
$G_0 = 3 + 2 + 1 = 6$

For episode 2 (starting from state A and taking action a):

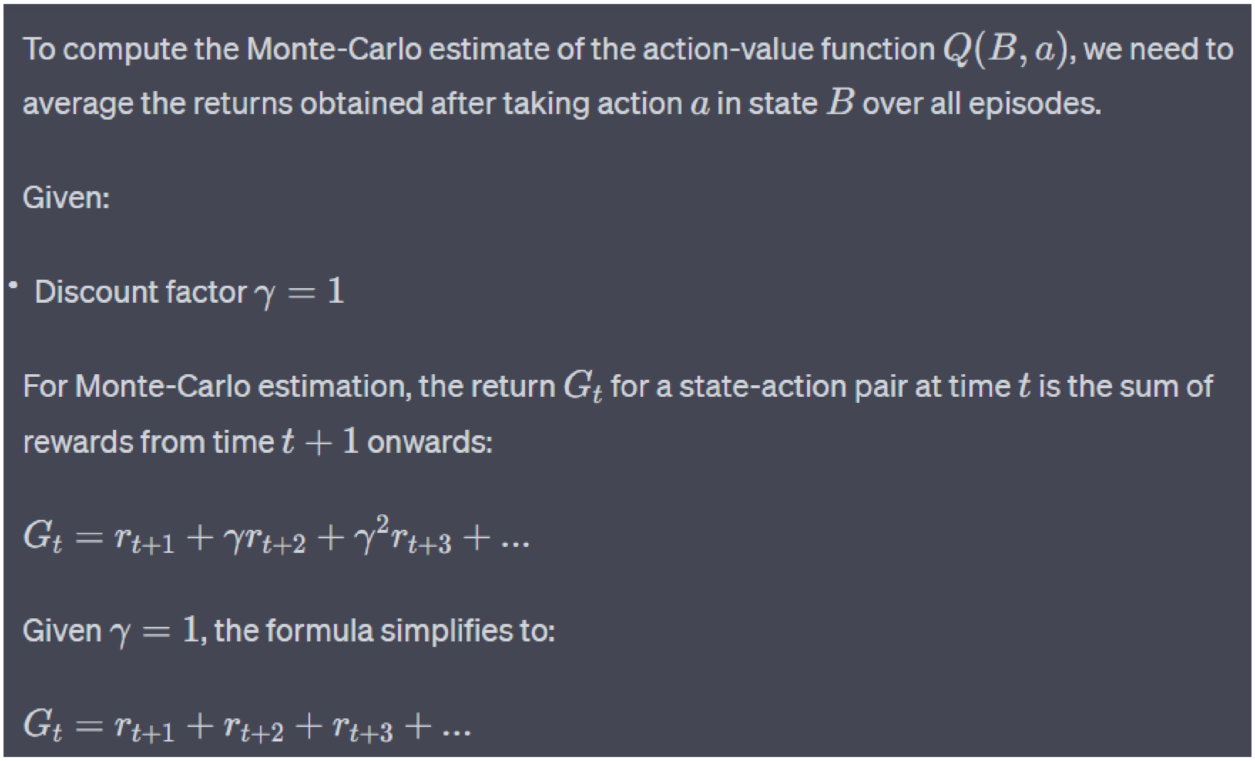
$G_0 = r_1 + r_2 + r_3$
$G_0 = 4 + 2 + 1 = 7$

Now, the Monte-Carlo estimate of $Q(A, a)$ is the average of the returns after taking action a in state A over the two episodes:

$Q(A, a) = \frac{G_0^{(1)} + G_0^{(2)}}{2}$
$Q(A, a) = \frac{6+7}{2}$
$Q(A, a) = 6.5$

So, the Monte-Carlo estimate of the action-value function $Q(A, a)$ is 6.5.

Question 8:



To compute the Monte-Carlo estimate of the action-value function $Q(B, a)$, we need to average the returns obtained after taking action a in state B over all episodes.

Given:

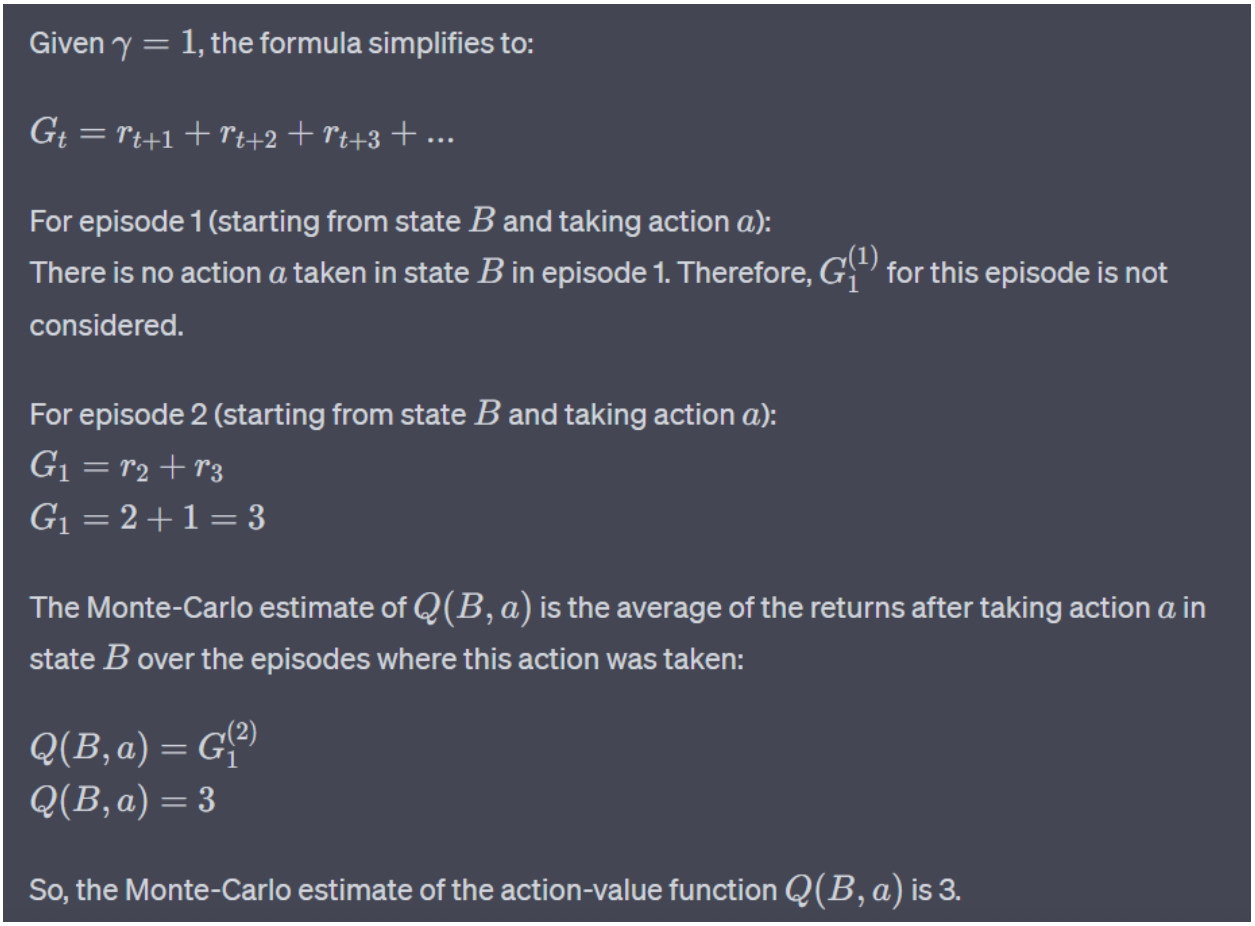
- Discount factor $\gamma = 1$

For Monte-Carlo estimation, the return G_t for a state-action pair at time t is the sum of rewards from time $t + 1$ onwards:

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots$$

Given $\gamma = 1$, the formula simplifies to:

$$G_t = r_{t+1} + r_{t+2} + r_{t+3} + \ldots$$



Given $\gamma = 1$, the formula simplifies to:

$$G_t = r_{t+1} + r_{t+2} + r_{t+3} + ...$$

For episode 1 (starting from state B and taking action a):
There is no action a taken in state B in episode 1. Therefore, $G_1^{(1)}$ for this episode is not considered.

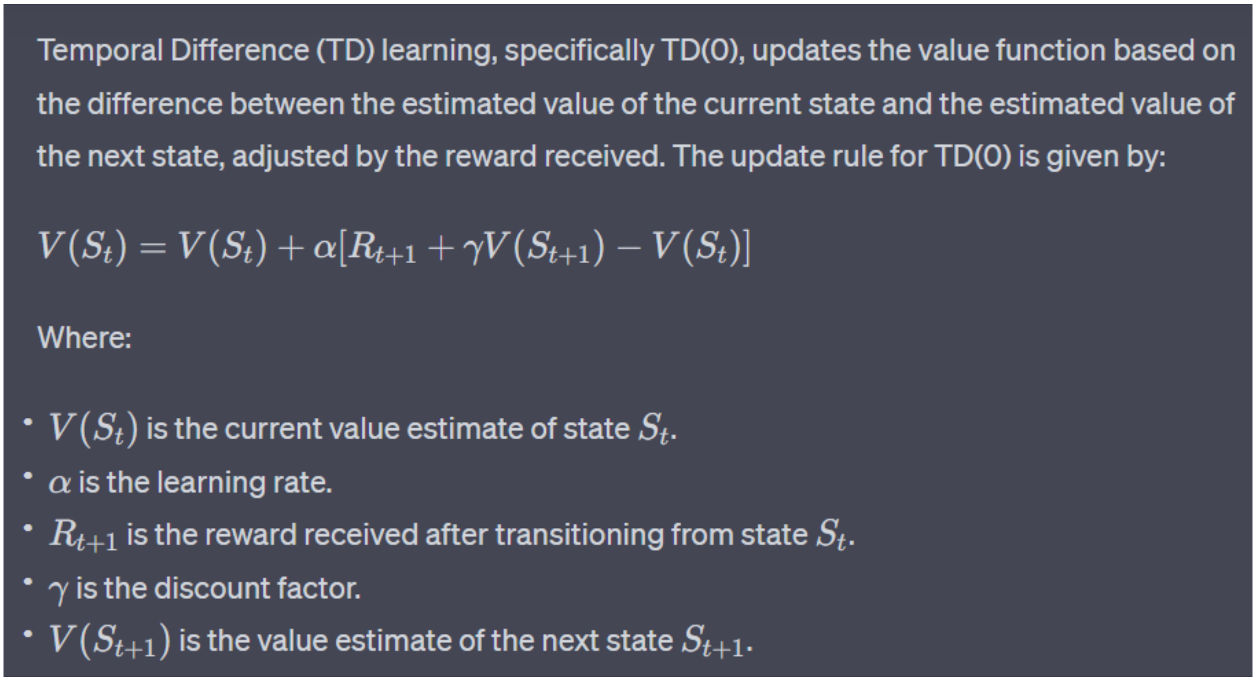
For episode 2 (starting from state B and taking action a):
$G_1 = r_2 + r_3$
$G_1 = 2 + 1 = 3$

The Monte-Carlo estimate of $Q(B, a)$ is the average of the returns after taking action a in state B over the episodes where this action was taken:

$$Q(B, a) = G_1^{(2)}$$
$$Q(B, a) = 3$$

So, the Monte-Carlo estimate of the action-value function $Q(B, a)$ is 3.

Question 9:

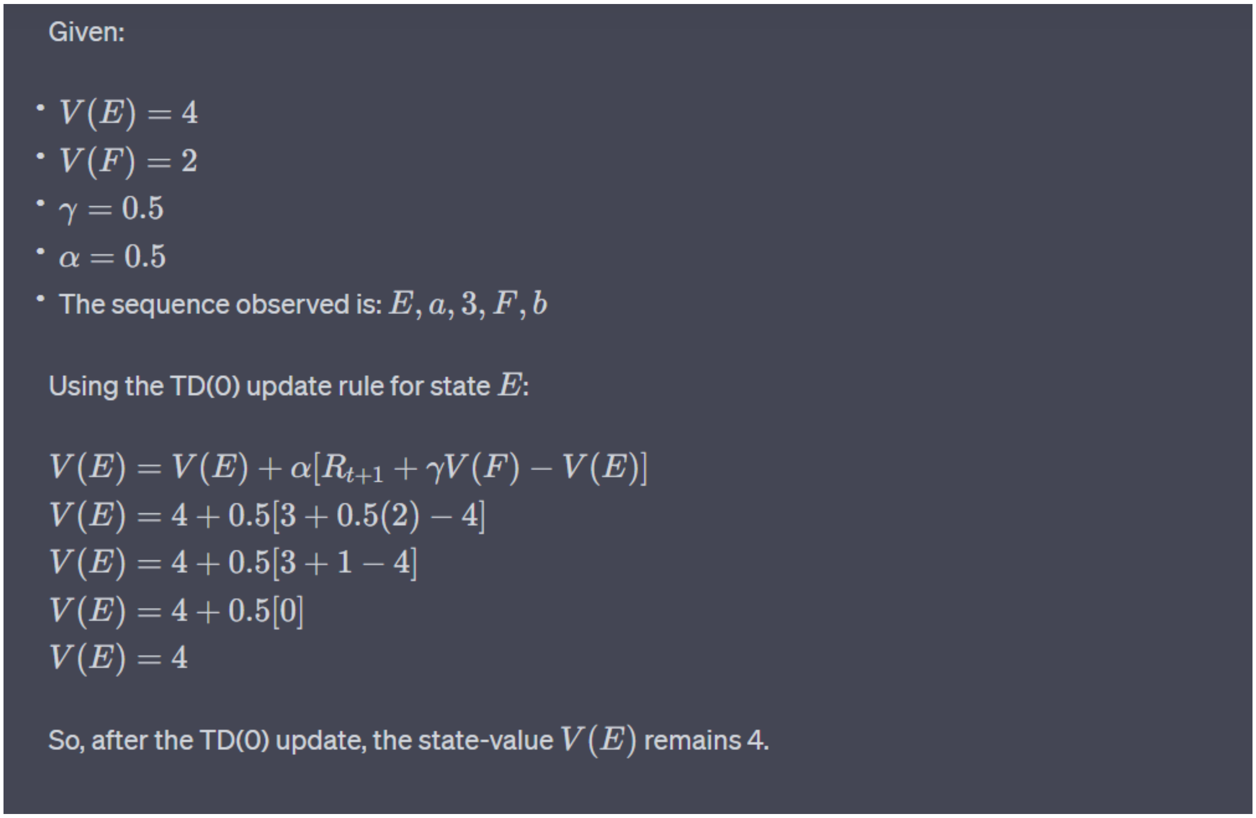


Temporal Difference (TD) learning, specifically TD(0), updates the value function based on the difference between the estimated value of the current state and the estimated value of the next state, adjusted by the reward received. The update rule for TD(0) is given by:

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

Where:

- $V(S_t)$ is the current value estimate of state S_t.
- α is the learning rate.
- R_{t+1} is the reward received after transitioning from state S_t.
- γ is the discount factor.
- $V(S_{t+1})$ is the value estimate of the next state S_{t+1}.



Given:

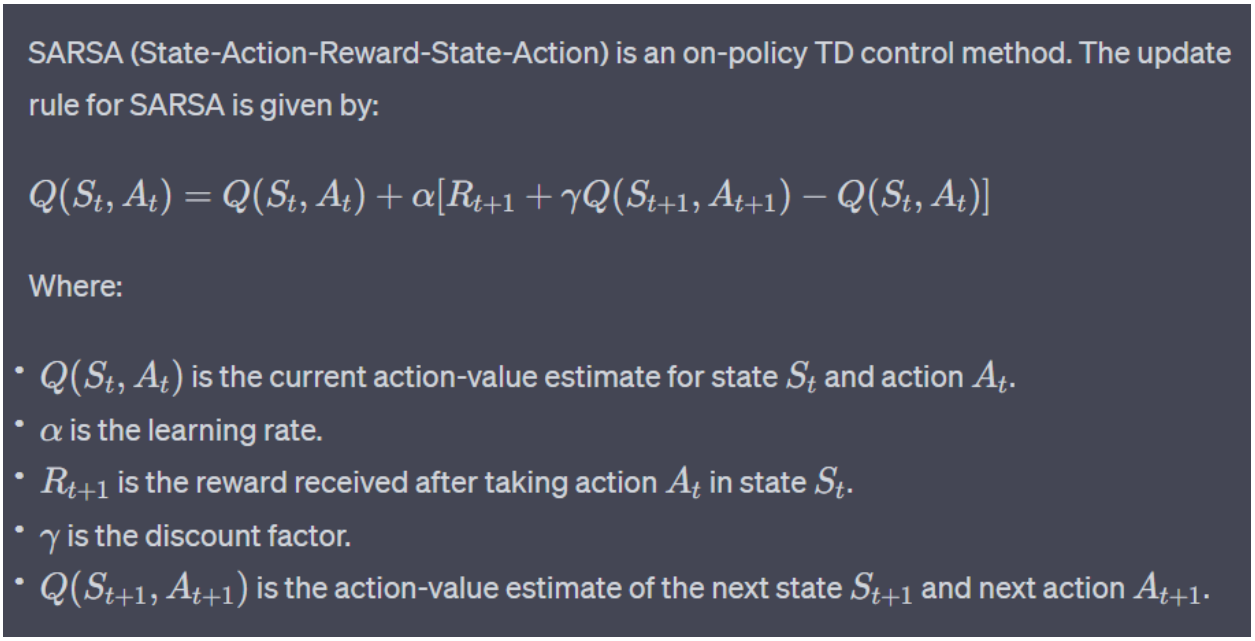
- $V(E) = 4$
- $V(F) = 2$
- $\gamma = 0.5$
- $\alpha = 0.5$
- The sequence observed is: $E, a, 3, F, b$

Using the TD(0) update rule for state E:

$$V(E) = V(E) + \alpha[R_{t+1} + \gamma V(F) - V(E)]$$
$$V(E) = 4 + 0.5[3 + 0.5(2) - 4]$$
$$V(E) = 4 + 0.5[3 + 1 - 4]$$
$$V(E) = 4 + 0.5[0]$$
$$V(E) = 4$$

So, after the TD(0) update, the state-value $V(E)$ remains 4.

Question 10:

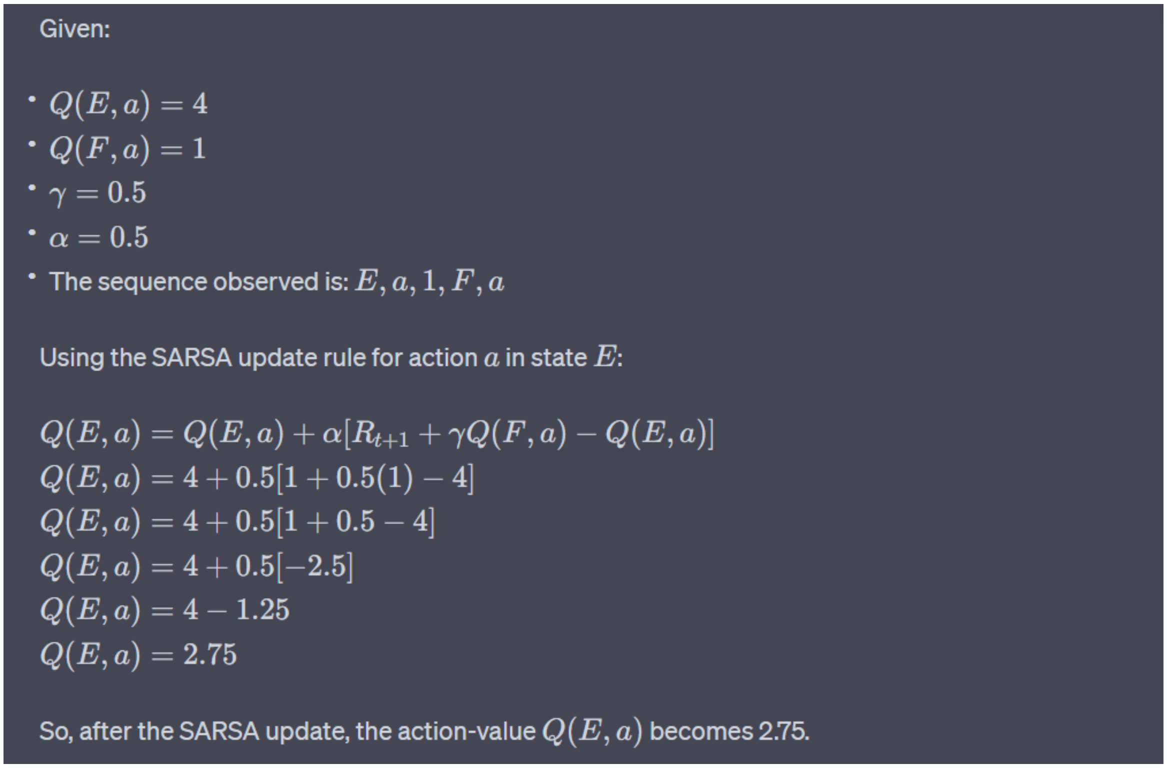


SARSA (State-Action-Reward-State-Action) is an on-policy TD control method. The update rule for SARSA is given by:

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

Where:

- $Q(S_t, A_t)$ is the current action-value estimate for state S_t and action A_t.
- α is the learning rate.
- R_{t+1} is the reward received after taking action A_t in state S_t.
- γ is the discount factor.
- $Q(S_{t+1}, A_{t+1})$ is the action-value estimate of the next state S_{t+1} and next action A_{t+1}.



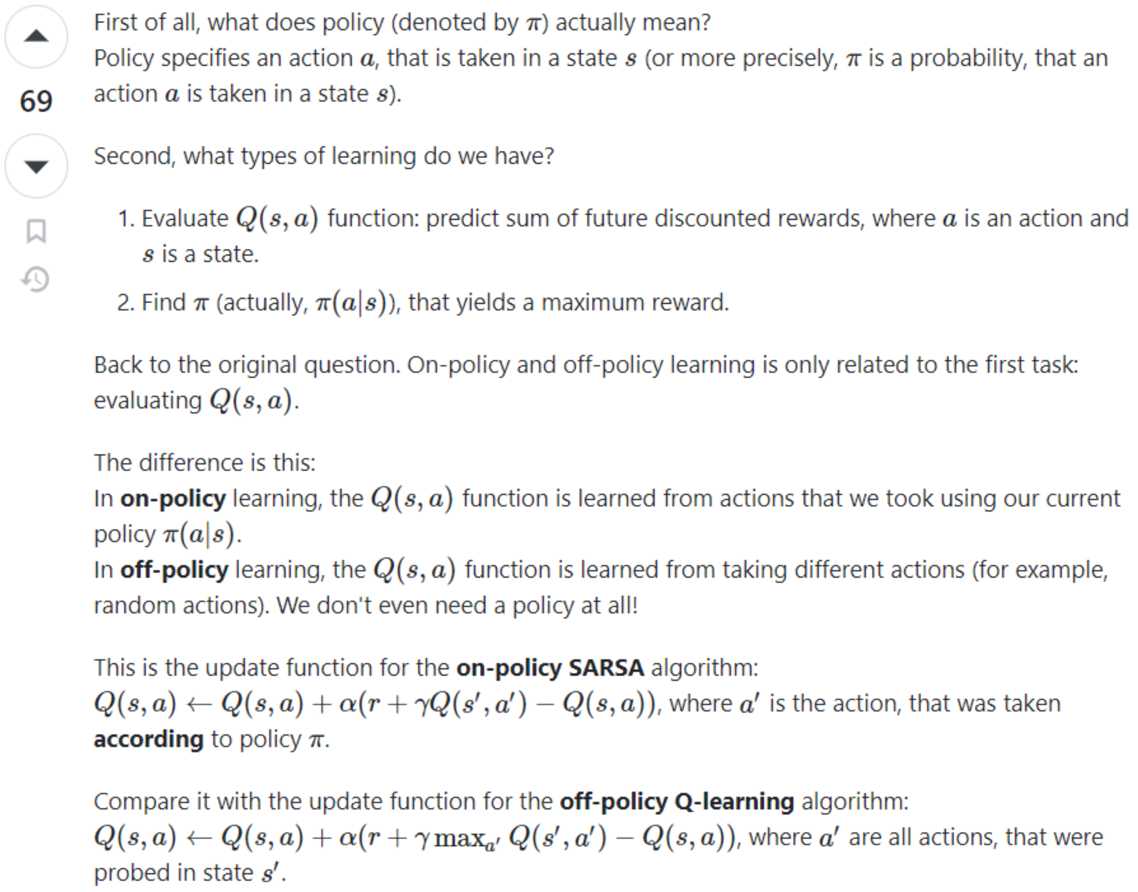
Given:

- $Q(E, a) = 4$
- $Q(F, a) = 1$
- $\gamma = 0.5$
- $\alpha = 0.5$
- The sequence observed is: $E, a, 1, F, a$

Using the SARSA update rule for action a in state E:

$$Q(E, a) = Q(E, a) + \alpha[R_{t+1} + \gamma Q(F, a) - Q(E, a)]$$
$$Q(E, a) = 4 + 0.5[1 + 0.5(1) - 4]$$
$$Q(E, a) = 4 + 0.5[1 + 0.5 - 4]$$
$$Q(E, a) = 4 + 0.5[-2.5]$$
$$Q(E, a) = 4 - 1.25$$
$$Q(E, a) = 2.75$$

So, after the SARSA update, the action-value $Q(E, a)$ becomes 2.75.



First of all, what does policy (denoted by π) actually mean?
Policy specifies an action a, that is taken in a state s (or more precisely, π is a probability, that an action a is taken in a state s).

Second, what types of learning do we have?

1. Evaluate $Q(s, a)$ function: predict sum of future discounted rewards, where a is an action and s is a state.
2. Find π (actually, $\pi(a|s)$), that yields a maximum reward.

Back to the original question. On-policy and off-policy learning is only related to the first task: evaluating $Q(s, a)$.

The difference is this:
In **on-policy** learning, the $Q(s, a)$ function is learned from actions that we took using our current policy $\pi(a|s)$.
In **off-policy** learning, the $Q(s, a)$ function is learned from taking different actions (for example, random actions). We don't even need a policy at all!

This is the update function for the **on-policy SARSA** algorithm:
$Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma Q(s', a') - Q(s, a))$, where a' is the action, that was taken **according** to policy π.

Compare it with the update function for the **off-policy Q-learning** algorithm:
$Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s, a))$, where a' are all actions, that were probed in state s'.