Question 1:

The correct answer is:
b. a mapping from state to action.

Question 2:

Given the definition of discounted returns from Sutton \& Barto's book, the return \(G_t \) at time \(t \) is defined as:

\[G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \]

Where:
\begin{itemize}
\item \(\gamma \) is the discount factor.
\item \(R_{t+1} \), \(R_{t+2} \), etc. are the rewards received at times \(t+1 \), \(t+2 \), etc.
\end{itemize}

Given the provided information:
\[\gamma = 0.5 \]
\[R_1 = 9 \]
\[R_2 = 14 \]
\[T = 2 \]

We want to find \(G_0 \). Since \(T = 2 \), the episode ends after 2 time steps. Thus, the return \(G_0 \) at time 0 is:

\[G_0 = R_1 + \gamma R_2 \]

Plugging in the given values:

\[G_0 = 9 + 0.5(14) \]
\[G_0 = 9 + 7 \]
\[G_0 = 16 \]

So, \(G_0 = 16 \).

Question 3:

The correct answer is:

True.

In the "Snakes and Ladders" board game, the next state (position on the board) depends only on the current state and the outcome of the dice roll, and not on the sequence of states that preceded it. This satisfies the Markov property.

Question 4:

To determine which action the agent will take, we need to compute the expected value of each action using the given value function and the discount factor.

For the left action:
Expected value = Immediate reward + (discount factor * value of the left state)
Expected value (left) = 0 + (0.5 * 4) = 2

For the right action:
Expected value = Immediate reward + (discount factor * value of the right state)
Expected value (right) = 0 + (0.5 * 8) = 4

Since the expected value for the right action (4) is greater than the expected value for the left action (2), the agent will choose the right action.

The correct answer is:

b. right.

Question 5:

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

[S1] - [S2] - [S3]
-8 ? 12
The agent is in the middle state (S2) with an unknown value. We need to compute the value of the middle state given a uniformly random policy.

For a uniformly random policy, the agent has a 0.5 probability of choosing either the left or the right action.

Value of S2 based on the left action:
Expected value (left) = Immediate reward + (discount factor * value of the left state)
Expected value (left) = 0 + (0.5 * -8) = -4

Value of S2 based on the right action:
Expected value (right) = Immediate reward + (discount factor * value of the right state)
Expected value (right) = 0 + (0.5 * 12) = 6

Given the uniformly random policy, the value of S2 is the average of the expected values of the left and right actions:

Value(S2) = 0.5 * Expected value (left) + 0.5 * Expected value (right)
Value(S2) = 0.5 * (-4) + 0.5 * 6
Value(S2) = -2 + 3
Value(S2) = 1

So, the value of the middle state (S2) is 1.

Question 6:

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

```
[S1] - [S2] - [S3]
  4      8      16
```

The agent is in the middle state (S2). We need to check if the value of the middle state represents an optimal policy using the Bellman optimality equation.

For the left action:
Expected value (left) = Immediate reward + (discount factor * value of the left state)
Expected value (left) = 0 + (0.5 * 4) = 2

For the right action:
Expected value (right) = Immediate reward + (discount factor * value of the right state)
Expected value (right) = 0 + (0.5 * 16) = 8

The Bellman optimality equation states that the value of a state under an optimal policy is the maximum expected return achievable by any action in that state.

Given the above calculations, the maximum expected return from the middle state (S2) is 8 (from moving right). However, the provided value for the middle state is also 8.

Since the value of the middle state matches the maximum expected return, it satisfies the Bellman optimality equation.

The correct answer is:

a. yes

Question 7:
Given the 3-element linear world:

-3, ?, 8

For action a:
Expected value a = 0 + 0.5 * (0.5 * (-3) + 0.5 * 8) = 1.25

For action b:
Expected value b = 0 + 0.5 * (1 * (-3) + 0 * 8) = -1.5

The updated value of the middle state after one step of value iteration is the maximum of the expected values for actions a and b:

Updated value for the middle state = max(1.25, -1.5) = 1.25

So, the updated value of the middle state after one step of value iteration is 1.25.

Question 8:

The correct answer is:

False.

An optimal state value function provides the maximum expected return from any given state. However, there can be multiple policies that achieve this optimal value. Especially in cases where multiple actions lead to the same expected return, there can be multiple optimal policies. Thus, an optimal state value function does not necessarily define a unique optimal policy.

Question 9:
The correct answer is:

True.

The optimal action value function, often denoted as Q* , represents the maximum expected return for taking a particular action in a particular state and then following the optimal policy thereafter. For a given task or environment, there is always a unique optimal action value function, even though there might be multiple optimal policies that achieve this value.

Question 10:
The correct answer is:

False.

While an optimal action value function, Q* provides the maximum expected return for taking a particular action in a given state and then following the optimal policy, it doesn't necessarily lead to a unique optimal policy. There can be situations where multiple actions have the same optimal value in a given state, leading to multiple optimal policies.