1.

**complete step by step derivation of Bellman Equation for state value function**

Sure, here's a step-by-step derivation of the Bellman equation for the state-value function:

1. The state-value function V(s) represents the expected total reward starting from state s and following a policy that selects actions according to the V-values.
2. The expected total reward starting from state s can be decomposed into the immediate reward R(s) and the expected future reward starting from the next state s':

V(s) = E[R(s) + γ V(s') | s]

where γ is the discount factor that determines the importance of future rewards.

1. The expected future reward starting from the next state s' can be expressed as the expected value of the state-value function over all possible next states:

V(s') = E[V(s'') | s']

where s'' is the next state after s'.

1. Substituting equation (3) into equation (2) gives:

V(s) = E[R(s) + γ E[V(s'') | s'] | s]

1. Applying the law of iterated expectations, we can simplify equation (4) to:

V(s) = E[R(s) + γ V(s') | s]

1. Therefore, we have derived the Bellman equation for the state-value function V(s) as:

V(s) = E[R(s) + γ V(s') | s]

This equation expresses the fact that the value of a state is equal to the immediate reward received plus the discounted value of the expected future reward starting from the next state. This equation forms the basis for many reinforcement learning algorithms such as value iteration and policy iteration.

**complete step by step derivation of Bellman Equation for action value function**

Sure, here's a step-by-step derivation of the Bellman equation for the action-value function:

1. The action-value function Q(s, a) represents the expected total reward starting from state s, taking action a, and following a policy that selects actions according to the Q-values.
2. The expected total reward starting from state s and taking action a can be decomposed into the immediate reward R(s, a) and the expected future reward starting from the next state s':

Q(s, a) = E[R(s, a) + γ V(s') | s, a]

where V(s') is the value function representing the expected total reward starting from state s'.

1. The value function V(s') can be expressed as the expected value of the action-value function over all possible actions:

V(s') = E[Q(s', a') | s']

where a' is the action taken in the next state s'.

1. Substituting equation (3) into equation (2) gives:

Q(s, a) = E[R(s, a) + γ E[Q(s', a') | s'] | s, a]

1. Applying the law of iterated expectations, we can simplify equation (4) to:

Q(s, a) = E[R(s, a) + γ max\_a' Q(s', a') | s, a]

where max\_a' Q(s', a') is the maximum expected future reward starting from the next state s' over all possible actions a'.

1. Therefore, we have derived the Bellman equation for the action-value function Q(s, a) as:

Q(s, a) = E[R(s, a) + γ max\_a' Q(s', a') | s, a]

This equation expresses the fact that the value of a state-action pair is equal to the immediate reward received plus the discounted value of the best action to take in the successor state.