
Multiple-Choice Questions (MCQs)

1. (Theoretical) Question: According to the slides, what is the definition of a "stationary policy"?

a) A policy that specifies a fixed sequence of actions, regardless of the state.

b) A policy that is a function of both the current state and the time step.

c) A policy that specifies what the agent should do as a function of only the current state.

d) A policy that always tells the agent to stay in its current state.

✅ **Correct Answer:** (c) **Explanation:** Slide 17 explicitly defines a stationary policy as one that is a function of the state. A non-stationary policy is a function of the state and the time.

Here are 5 multiple-choice questions (2 numerical, 3 theoretical) and 2 numerical short-answer questions based on your request.

---

2. (Numerical) Question: In the 3×4 grid world (diagram on slide 10), the robot is in state s14 and tries to move 'right'. What is the probability that the robot stays in state s14?

	1	2	3	4
1	Start			
2		X		-1
3				+1

a) 0.1

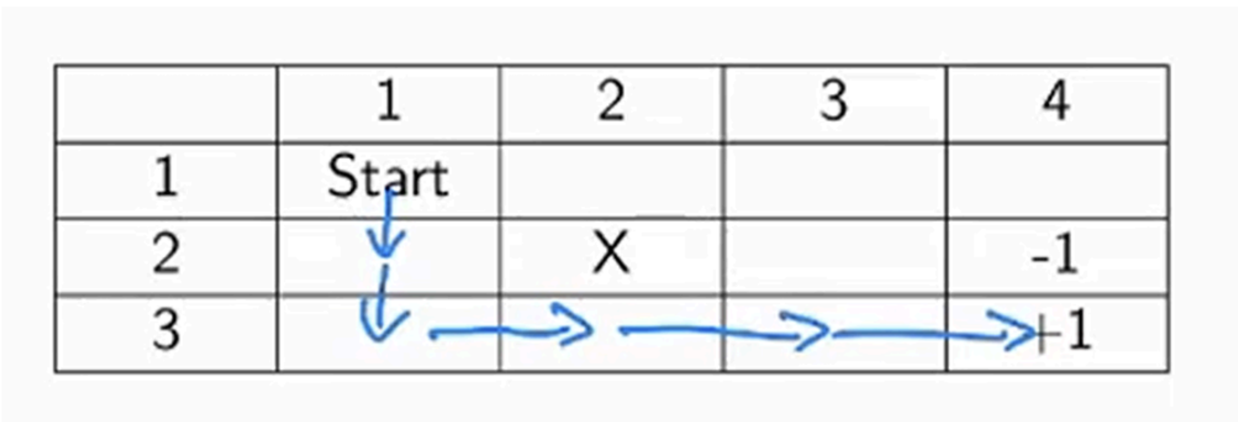
b) 0.8

c) 0.9

d) 1.0

✅ **Correct Answer:** (c)   **Explanation:** The transition model (slide 9) states an action has a 0.8 probability of its intended effect, 0.1 of a 90-deg left turn, and 0.1 of a 90-deg right turn.

- **Intended ('right'):** Bumps the wall, stays in s14 (Prob: 0.8).
- **90-deg left ('up'):** Bumps the wall, stays in s14 (Prob: 0.1).
- **90-deg right ('down'):** Moves to s24 (Prob: 0.1).
- The total probability of staying in s14 is 0.8+0.1=0.9.



	1	2	3	4
1	Start			
2		X		-1
3				+1

3. (Theoretical) Question: The Policy Iteration algorithm is described as alternating between two main steps. What are these two steps?

a) Value Iteration and Policy Improvement

b) Policy Evaluation and Policy Improvement

c) Synchronous Updates and Asynchronous Updates

d) Bellman Equations and Q-Value Calculation

✅ **Correct Answer:** (b)   **Explanation:** The slides state that Policy Iteration alternates between two steps: (1) **Policy Evaluation**, to calculate the utility Vπi(s) for the current policy, and (2) **Policy Improvement**, to calculate a new policy πi+1 using those utilities .

4. (Numerical) Question: In the Value Iteration example for the 3×4 grid, the algorithm starts with $V0(s)=0$ for all non-goal states (see V0 table on slide 43). Given $\gamma=1$ and $R(s)=-0.04$ for non-goals, what is the value of $V1(s33)$?

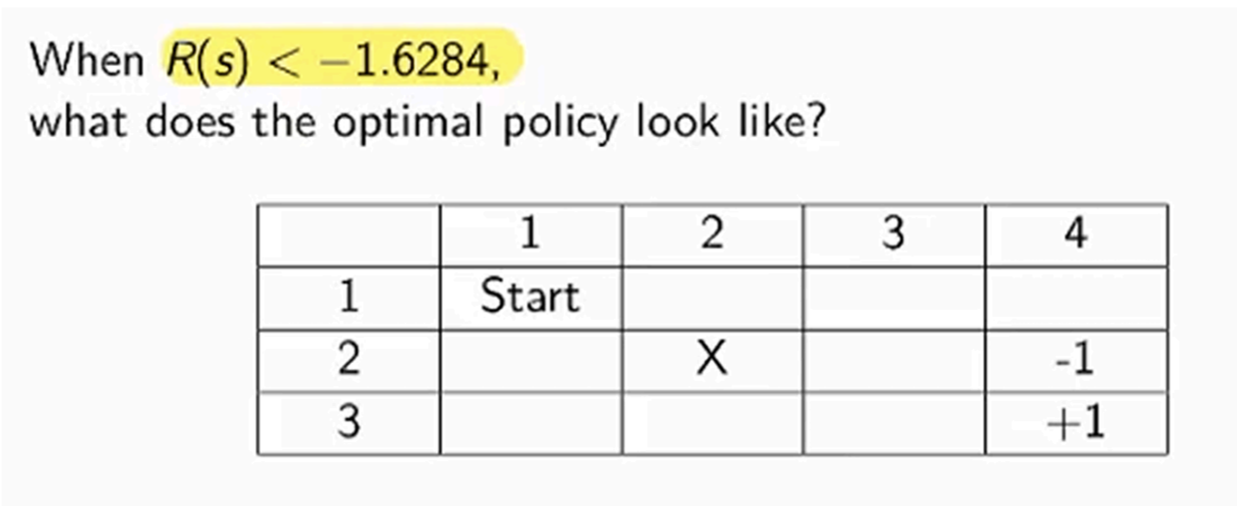
a) -0.04 b) 0.80 c) 0.76 d) 1.00

✅ **Correct Answer:** (c) **Explanation:** The update rule is $V_{i+1}(s) \leftarrow R(s)+\gamma \max_a \sum P(s' \mid s,a) V_i(s')$. For $V1(s33)$:

- $R(s33)=-0.04$.
- The best action 'a' from s33 is 'right', which has a 0.8 probability of reaching s34 (value $V0(s34)=+1$).
- The max expected utility from V0 is $\approx (0.8 \times V0(s34)) = 0.8 \times 1 = 0.8$.
- Therefore, $V1(s33) = -0.04 + (1 \times 0.8) = 0.76$.

—---

5. (Theoretical) Question: Why does the optimal policy for the 3×4 grid world (slide 19) depend on the value of $R(s)$for non-goal states?


When $R(s) < -1.6284$,
what does the optimal policy look like?

	1	2	3	4
1	Start			
2		X		-1
3				+1

a) Because $R(s)$ determines the location of the wall at s22.
b) Because the agent's policy must balance the "cost of living" (the reward $R(s)$ for each step) against the final reward at a goal state.

c) Because the transition probabilities $P(s' \mid s,a)$ change when $R(s)$ changes.
d) Because a positive $R(s)$ makes the discount factor γnegative.

✅ **Correct Answer:** (b) **Explanation:** The optimal policy is a trade-off. When $R(s)$ is very negative (e.g., "painful," $R(s)<-1.6284$), the agent wants to exit *fast*, even to the -1 goal. When $R(s)$ is only slightly negative (e.g., "slightly dreary"), the agent is willing to take a *safer*, longer path to get the +1 goal.

Q.1

In the 3×4 grid world, the Value Iteration algorithm is applied with $\gamma=1$ and $R(s)=-0.04$ for non-goal states. You are given the initial values $V0(s)$ from the table on **slide 43**, where all non-goal states are 0, $V0(s24)=-1$, and $V0(s34)=+1$.

Calculate the value V1(s23).

	1	2	3	4
1	0	0	0	0
2	0	X	0	-1
3	0	0	0	+1

Answer: We use the Value Iteration update rule shown on **slide 42** and **slide 44**: $V1(s23)=R(s23)+\gamma \max_a \sum P(s' \mid s23,a)V0(s')$ Given $R(s23)=-0.04$ and $\gamma=1$. We must check the expected utility for all 4 actions using the $V0$values:

1. **Action 'up':** $0.8\times V0(s13)+0.1\times V0(s22\rightarrow s23)+0.1\times V0(s24)$
$=(0.8\times 0)+(0.1\times 0)+(0.1\times -1)=-0.1$

2. **Action 'down':** $0.8\times V0(s33)+0.1\times V0(s22\rightarrow s23)+0.1\times V0(s24)$
$=(0.8\times 0)+(0.1\times 0)+(0.1\times -1)=-0.1$

3. **Action 'left':** $0.8\times V0(s22\rightarrow s23)+0.1\times V0(s33)+0.1\times V0(s13)$
$=(0.8\times 0)+(0.1\times 0)+(0.1\times 0)=0$

4. **Action 'right':** $0.8\times V0(s24)+0.1\times V0(s33)+0.1\times V0(s13)$
$=(0.8\times -1)+(0.1\times 0)+(0.1\times 0)=-0.8$

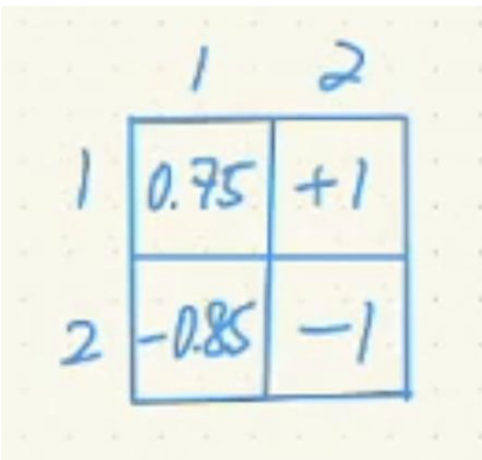
The max of these outcomes is 0. $V1(s23)=-0.04+(1\times 0)=-0.04$.

—---

Question 2

In the 2×2 grid Policy Iteration example (diagram on **slide 61**), the values after Policy Evaluation are V(s11)=0.75, V(s21)=−0.85, V(s12)=+1, and V(s22)=−1. The transition model is 0.8 for the intended direction, 0.1 for a 90-degree left turn, and 0.1 for a 90-degree right turn.

Calculate the expected utility (Q-value) for Q(s11,right) during the Policy Improvement step.



Answer: We calculate the Q-value as the sum of (probability × utility) for all possible outcomes, as shown on **slide 61**: Q(s11,right)=P(s′∣s11,right)V(s′)

1. **Intended ('right'):** Moves to s12. Probability = 0.8. Utility = 0.8×V(s12)=0.8×(+1)=0.8
2. **90-deg left ('up'):** Bumps wall, stays in s11. Probability = 0.1. Utility = 0.1×V(s11)=0.1×(0.75)=0.075
3. **90-deg right ('down'):** Moves to s21. Probability = 0.1. Utility = 0.1×V(s21)=0.1×(−0.85)=−0.085

Total Q-value: 0.8+0.075−0.085=0.79.