# Reinforcement Learning

## Direct MCQs for End Course Test

**Question 1:** In a Game with a grid size of 84 X 84 with close to 7 million possible states, if we want the Agent to learn the best policy through exploration, which is the best algorithm to use?

1. Actor Critic
2. Policy Gradient Using REINFORCE
3. Q Learning
4. Deep Q Learning

**Answer:** Option 3. Since we want the Agent to learn using Exploration and it is not mentioned that we will be using images to train the Agent, Q Learning is the best Algorithm to use because it will be faster in converging due to Model Free Approach.

**Question 2:** Assume that we need the Agent to learn the uncertainties in an environment and commit less errors and maximize the reward. Which of the following options is the correct choice for such situations?

1. Dynamic Programming
2. Monte Carlo Control Methods
3. Value Iteration
4. Temporal Difference Learning

**Answer:** Option 4. Temporal Difference combines Dynamic Programming and Monte Carlo approaches because TD methods do not require a model of the environment, of its reward and next state probability distributions, there is no need to wait until the end of an episode and learn from each transition, irrespective of what subsequent actions are taken. Hence, TD Approach is the correct choice.

**Question 3:** There can only be one local maxima and one global maxima when using Policy Gradients.

1. True. Because the agent will explore the entire solution space and finalize the policies.
2. False. Because the agent will only choose those policies that does not end in negative rewards or penalties.
3. True. Because the agent’s objective is to find only one global maxima which includes one local maxima.
4. False. Because it depends upon the sampling efficiency in importance sampling and it is possible that there is more than one local maxima.

**Answer 3:** False. It depends upon the sampling efficiency in importance sampling. Depending upon samples generated, there can be more than one local maxima. However, the global maxima remains singular.

**Question 4:** Temporal Difference (TD) Learning follows the bootstrapping approach to update the action – value function whereas Monte Carlo averages out the function to stability. In which scenario, Monte Carlo is better than Temporal Difference Learning?

1. Monte Carlo will outperform TD Learning when off – policy methods are used as function approximators because there is a chance of non – convergence.
2. Monte Carlo is a better choice when performance assessment of multiple agents is required using policy evaluation.
3. TD Learning is always better than Monte Carlo i.e. there is no such scenario because Monte Carlo is always slower to converge than TD Learning and it has higher variance than TD Learning
4. Monte Carlo is preferred when the rules of the environment are fixed

**Answer:** Both 1 and 2. The bootstrapping process in TD Learning updates a function Q(s,a) on the next value Q(s',a') using the current estimates. In the beginning, the estimates do not contain any information from any real rewards or any real transitions. This can create bias. If the learning goes well, the bias will reduce asymptotically else the convergence will never happen.

**Question 5:** Which Algorithm places importance on preferences of actions leading to rewards?

1. Upper Confidence Bound Selection
2. Gradient Bandit
3. Epsilon Greedy Selection
4. Contextual Bandits

**Answer:** Option 2. In Gradient Bandit Algorithm, respective preference of one action over another is important which results in rewards. Preference does not directly relate to rewards, but it causes the Agent to choose actions that produce rewards and thereby more preference is provided to that action.

**Question 6:** If there is a complex reward function and a combination of Algorithms (both model based and model free) and still the learning is not converging then what could be the most likely solution?

1. Add more training data
2. It is a problem of sparse rewards and so reward shaping will resolve this issue
3. Bootstrapping via Temporal Difference methods should be implemented
4. Check quality of training data, set a benchmark and stop the training if the Agent is not learning anything new

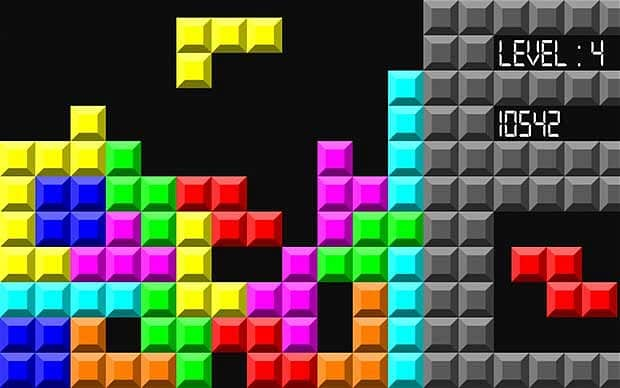
**Answer:** Option 4. The Agent learns what it is shown. So if everything else fails, then the only logical option is to check the quality of the training data because RL follows GIGO rule. “Garbage In – Garbage Out”.

**Question 7:** Can the entire environment be modeled before an Agent starts exploring? If yes, which Algorithm does this?

1. No, only the next state – action pair can be estimated
2. Yes, Monte Carlo & SARSA
3. Yes, Monte Carlo & Q – Learning
4. Yes, Monte Carlo

**Answer:** Option 3. Since the environment can be stochastic, Monte Carlo will be the best suited Algorithm to estimate the environment while Q – Learning can help explore those estimations and update the Q – Matrix as it is an off – policy learner.

**Question 8:** Refer to the famous game “Tetris” as described below.



This is a classic arcade game where blocks of different shapes fall randomly and users have to place them strategically to clear the level and score points. If there are no gaps in the parts and the entire line is filled with blocks, the line disappears leading to reduction in the height of the blocks which gives the user points.

The aim is to clear as many levels as possible. The speed of the blocks falling, increases with each advancement in level.

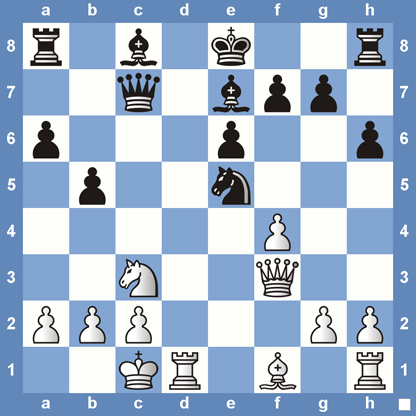
Training an RL Agent on the famous game of “Tetris” will be most accurate by which of the following options?

1. Recreation of the Tetris world in grid world and training the RL Agent using any of the Exploration algorithms
2. Providing gaming screens of highest scored human players and using DQN to train the Agent
3. Defining the rules and using SARSA to find Optimal Policy
4. Using Monte Carlo and Value Iteration to train the Agent

**Answer:** Option 3. The game of Tetris has limited rules of the environment that can easily be defined. SARSA can be used to find the Optimal Policy to maximize scores rather than Value Iteration since Value Iteration will promote more Exploration. Here we need to Exploit rather than Explore. DQN will not be used because the environment is not stochastic and has very few random actions that an Agent can take.

**Question 9:** Let us take the example of 4 games.

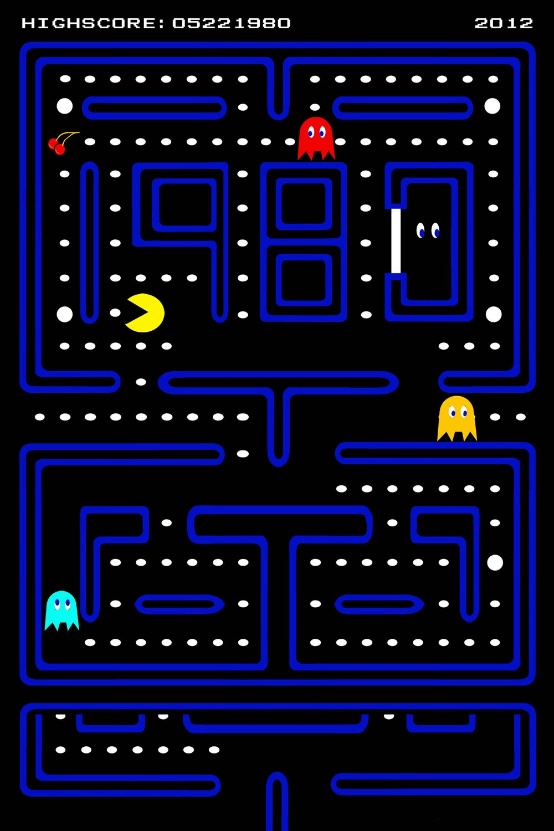
1. Chess – strategic board game played on chequered board with pieces like rook, knight, bishop, queen and pawns - aiming at checkmate of the opponent’s king.



1. Mario – a classical video game where the character has to clear various levels in order to save the princess.



1. Pacman – a classical arcade game wherein a character (the name is Pacman) is enclosed in a maze and is required to consume all the pellets while avoiding the ghosts or villains of the game.



1. Crazy Taxi – an arcade game from 1999 that is aimed at picking up the customers and dropping them off to their destination points as fast as possible.



Consider the set of games and the approaches as listed below:

1. Chess 1. Deep Q Networks
2. Mario 2. Epsilon Greedy
3. Pacman 3. Actor Critic
4. Crazy Taxi 4. Dynamic Programming

Which of the following option is the correct match?

* + - 1. A – 1, B – 2, C – 3, D – 4
      2. A – 4, B – 3, C – 2, D – 1
      3. A – 1, B – 3, C – 2, D – 4
      4. A – 4, B – 2, C – 1, D – 3

**Answer:** Option 2. The answer is based on how stochastic the environment is for each of the games. Mario has the most stochastic environment and hence Actor Critic is the right choice. In Pacman, the Agent can be greedy about eating the most dots, hence Epsilon Greedy is the right choice. Chess is directly solved by Dynamic Programming and Crazy Taxi will be based on Deep Q Networks as the number of actions to be taken in an environment is limited and can be resolved by Deep Q Networks.