DRAFT

Birla Institute of Technology & Science, Pilani
Work Integrated Learning Programmes Division
Second Semester 2022-2023
Mid-Semester Test (EC-2 Regular)

Course No. : AIMLCZG512 **Course Title**: Deep Reinforcement Learning
Nature of Exam : Closed Book **Weightage** : 30% **No. of Pages** = 3 ; **No. Of Questions =** 6;
Duration : 2 Hours; **Date of Exam**: 29-June-2025 - EN

Note to Students:

1. Answer all the questions.
2. Write your name and sign at the end of all the pages.
3. Assumptions made if any, should be stated clearly at the beginning of your answer.

NOTE: This solution is our first draft, might have some errors, corrected later. Verify if the solutions given is correct and provide correct answers if there are mistakes. Thank you

Question -1 [2 + 2 + 1 = 5 Marks]

A startup news platform wants to recommend one of 4 news categories (Tech, Sports, Politics, Entertainment) at each session using a bandit algorithm. Below are the results of 8 recommendations and the responses from the user. [Note: Use +1 as the reward if a click is received, 0 if a click is not received.]

Time Step	Category Chosen	Click Received? (1 = Yes, 0 = No)
1	Tech	1
2	Sports	0
3	Politics	1
4	Entertainment	1
5	Tech	0
6	Sports	1
7	Politics	1
8	Sports	1

(a) Using UCB with c=1.5, identify the next action to select. Show your calculations. [2 Marks]
(b) What is the ε-greedy action at the end of time step 8 if ε = 0.2? Show how the action is decided. [2 Marks]
(c) A new feature is introduced in the app where the recommendation system must consider the sequence of previously viewed articles by a user and the user's mood inferred from

interaction patterns to generate the next recommendation. Can this setting be effectively modelled using a multi-armed bandit approach? Justify your answer in no more than two well-reasoned statements. [1 mark]

Answer Key for Question 1:

(a) Using UCB with c = 1.5

Note: It is expected that the students understand UCB is an action selection mechanism, the returns are still computed using sample averaging. The answer is organized in 2 steps, in which the first step summarized the table given and computes Q(a) for each action. Students may skip this step if these calculations are computed as required for UCB(a).

Step 1: Summarize the info in the given scenario.

Count of selections and rewards for each category:
Tech - Count: 2, Total Rewards Received = 1, Q = 0.5
Sports - Count: 3, Total Rewards Received = 2, Q ≈ 0.6667
Politics - Count: 2, Total Rewards Received = 2, Q = 1.0
Entertainment - Count: 1, Total Rewards Received = 1, Q = 1.0

Step 2: Compute UCB values using the formula:
UCB(a) = Q(a) + c * sqrt(ln(t) / N(a))
where t = 9, c = 1.5, ln(9) ≈ 2.197

UCB(Tech): 0.5 + 1.5 * sqrt(2.197 / 2) ≈ 2.072
UCB(Sports): 0.6667 + 1.5 * sqrt(2.197 / 3) ≈ 1.949
UCB(Politics): 1.0 + 1.5 * sqrt(2.197 / 2) ≈ 2.572
UCB(Entertainment): 1.0 + 1.5 * sqrt(2.197) ≈ 3.223

The UCB(entertainment) is highest and hence the net action selected is Entertainment

(b) ε-greedy Action Selection (ε = 0.2)

With ε = 0.2, there is an 80% chance to exploit and 20% to explore.
Best Q-values: Politics = 1.0, Entertainment = 1.0 (tie)

80%: Pick randomly between Politics and Entertainment [Tie to be broken randomly]

20%: Randomly choose among all 4 actions

Likely ε-greedy action: Politics or Entertainment [Note, if the answer is either of these, with the reasoning its fine] [Note: if this reasoning is explicit, award marks, irrespective of approach]

(c) Modelling the Sequential Setting

No. The new setting depends on sequential context and user mood, making it unsuitable for standard multi-armed bandits. This means, we need to track the state (covers mood and other details) of the agent and learn to act optimally in each state. This means, we cannot model this scenario using Multi Armed Bandit formulation

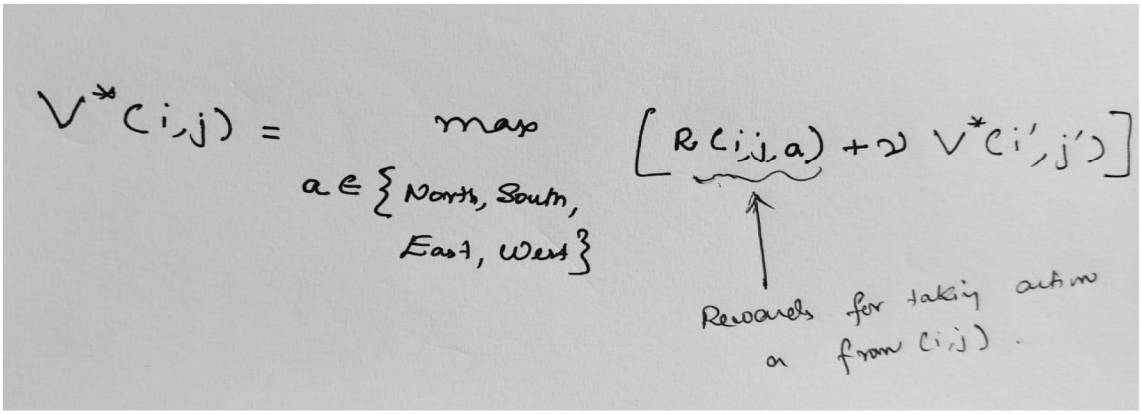
Question -2 [4 + 1 = 5 Marks]

Consider a robot navigating a 3×3 grid world. The robot can move in one of four directions: North, South, East, or West. Movements are **deterministic** unless blocked by the grid boundaries. If the robot attempts to move off the grid, it remains in the same cell and receives a reward of **–1**. The goal is located at **cell (3,3)** and provides a reward of **+10** upon entry. All other moves yield a reward of **–1**. Use a discount factor **γ = 0.9**.

(a) Write the Bellman optimal equation used for value iteration for any cell (i,j). [1.0 Marks]
(b) Perform one iteration of value update for all the states using the equation in (a), assuming initial V(s) = 0 for all states. Show the value table for all the states before and after the iteration [3.0 Marks]
(c) What would happen to the optimal policy if γ is reduced to 0.1? Explain briefly. [1.0 Marks]

Answer Key for Question 2:

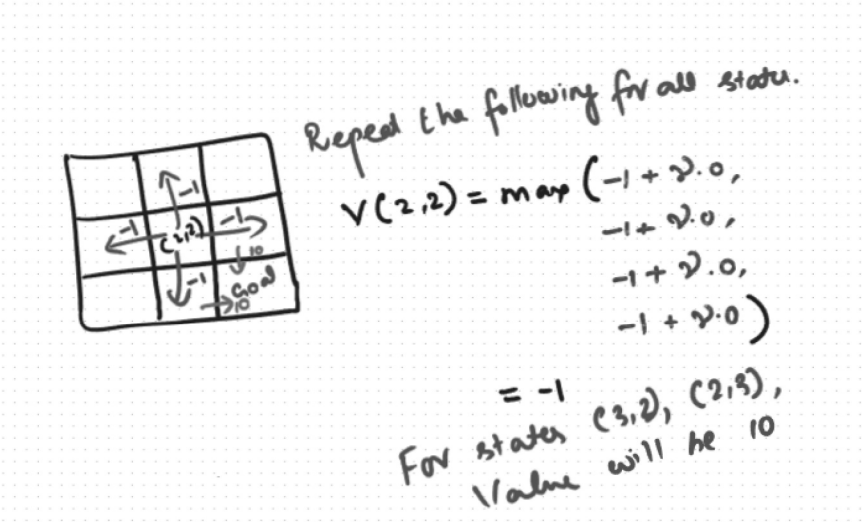
(a) Write the Bellman optimality equation, replacing the state as being in location (i, i). The transition probability should consider the deterministic nature of p(..., ….) given in the scenario. ;



Note: (1) The Expected expression is in the same line as shown here. Each term in the expression needs explanation.

(2) Expressions copied and pasted as is without making is suitable for the scenario will get 0 marks.

(b) Assume each cell (state) has the value initialised to 0.



As you repeat the above for all the states, the values for each state will be as below.

Cell	Max Value	Action Values (for each of 4 direction)
(1,1)	-1	[-1.0, -1.0, -1.0, -1.0]
(1,2)	-1	[-1.0, -1.0, -1.0, -1.0]
(1,3)	-1	[-1.0, -1.0, -1.0, -1.0]
(2,1)	-1	[-1.0, -1.0, -1.0, -1.0]
(2,2)	-1	[-1.0, -1.0, -1.0, -1.0]
(2,3)	10	[-1.0, 10.0, -1.0, -1.0]
(3,1)	-1	[-1.0, -1.0, -1.0, -1.0]
(3,2)	10	[-1.0, -1.0, 10.0, -1.0]

(c) The agent will discount the future rewards heavily, with the emphasis placed largely on immediate rewards. This, in a practical sense results in a situation where the agent takes many iterations to discover longer paths leading to the goal.

Question -3 [3 + 2 = 5 Marks]

(a) An intelligent traffic controller is designed to manage a four-way urban intersection. At every decision point, the controller observes the traffic density on each of the four incoming roads, which can be classified as **Low**, **Medium**, or **High**. Additionally, it receives inputs from pedestrian crossing buttons (Pressed or Not Pressed). Based on this information, the controller must select one of the following actions at each time step:

- **A1**: Allow **North-South** traffic
- **A2**: Allow **East-West** traffic
- **A3**: Allow **Pedestrians** to cross

The objective is to **minimise vehicle waiting time** and **ensure pedestrian safety** over time. Model the above system as a **Markov Decision Process (MDP)**. Provide reasoning for each of your choices. [3.0 Marks]

(b) A warehouse robot learns to navigate and pick items efficiently using reinforcement learning. Name one model-based and one model-free algorithm suitable for this task.

[1.0 Marks]. Describe one scenario where each approach (model-based and model-free) is more appropriate. [1.0 Marks]

Answer Key for Question 3:

a)

(1 m) State Definition & Action Space - Must include individual representation of every states

(1 m) Reward : Must be a inverse(or negative) function of the wait time + some function of pedestrian safety

(1 M) Sample dynamics illustration for the use case

MDP Components	Design Assumptions (There is a centralized pedestrian command input which results in all lanes to stop the vehicle)

State
Let
N=North Lane
S = South Lane
E = East Lane
W – West Lane
	{
	<N-Density, S-Density, E-Density, W-Density, CommonPedestrian>
	}
	
	Density ∈ {Low, Medium, High}
	Pedestrian ∈ {Pressed, Not-Pressed}

Action Space	{ A1, A2, A3} A1: Allow North-South traffic A2: Allow East-West traffic A3: Allow Pedestrians to cross
Reward	Positive Reward when pedestrian are allowed to cross -Total Vehicle Waiting Time Observed
State Transition	Eg., P(S,A,S`} = P ( <low, low, high, medium, Not-pressed>, A2, <low, medium, medium, medium, pressed> ) =0.6  Eg., r(S, A, S`) =0-20 vehicles waiting =-20 Lower the value of reward the better.

b)

(1m) Availability of the Model is the key.

(1m) One scenario w.r.t warehouse bot is expected.

Model based solution Dynamic Programming is ideal when the environment is fully known (transition model and rewards), allowing for iterative updates of the value function and policy. In warehouse robot a static environment or dynamic environment with fixed number of variables influencing the working of the robot is known then dynamic can be modelled as DP

Model-Free based solution Monte Carlo technique is suitable more when the environment is not fully known, as it estimates the value of a policy through sampling episodes without needing a model. In warehouse robot environments, due to more environment dynamics, non-stationary set ups, model free approach is preferred.

Question -4 [1.5 + 1.5 + 1 + 1 = 5 Marks]

A news recommendation system that recommends movies follows an **on-policy ε-soft strategy π** with ε = 0.2. Under this policy, it selects between two strategies with the following action probabilities:

- **π(Trending) = 0.5, π(Recent) = 0.5**

The system generates the following **4 episodes** (states are omitted since actions are taken directly):

- **Episode 1**: Trending (+3) → Recent (+4) → Trending (+5) → Recent (+7)
- **Episode 2**: Recent (+2) → Trending (+6) → Trending (+4)
- **Episode 3**: Trending (+1) → Recent (+3) → Recent (+5)
- **Episode 4**: Recent (+6) → Trending (+3) → Trending (+5) → Recent (+2)

Assume the **discount factor γ = 0.9**.

(a) Estimate the **state-value V(Trending)** using **First-Visit Monte Carlo estimation** from the above episodes. Show your calculation steps. [1.5 Marks]
(b) Estimate the **state-value V(Recent)** using **First-Visit Monte Carlo estimation**. Show your calculation steps. [1.5 Marks]
(c) Based on the estimated values, describe how you would update the **policy π** under **ε-soft on-policy improvement** with ε = 0.2. Specify the new action probabilities. [1 Mark]
(d) Write down 2 major issues of estimating the value function this way to learn a policy. [1 Mark]

Answer Key for Question 4:

a) First Visit MC : Value of "Recent"

Eg., In Episode 1 $4(0.9)^0+5(0.9)^1+7(0.9)^2= 14.17$
Value = 14.17

Episode 1 : 14.17
Episode 2 : 10.64
Episode 3 : 7.5
Episode 4 : 14.21
Average = ~11.63

b) First Visit MC : Value of "Trending"

Eg.. In Episode 1 : $3 + 0.9(4(0.9)^0+5(0.9)^1+7(0.9)^2)$
$=3+0.9(14.17)$
Value = 15.76

Episode 1 : 15.75
Episode 2 : 9.6
Episode 3 : 7.75
Episode 4 : 9.12
Average = ~10.56

c)

Greedy Action as per above = "Recent"
$P(Recent)= (1- \varepsilon) + \varepsilon / 2 = 0.8+0.1 = 0.9$
$P(Trending) = \varepsilon / 2 = 0.1$

d)

1) Off policy is sample inefficient and there is no reuse of trajectory(episode)
2) Off policy based learning process is affected by high variance. Noises or outlying experience may largely influence the value estimates. Sometimes this might affect the convergence of the algorithms

Question -5 [3 +2 = 5 Marks]

(a) Compare the following strategies used in RL: (i) ε-greedy (ii) Optimistic initial values (iii) Upper Confidence Bound (UCB). Provide one advantage and one drawback for each. [3 Marks]
(b) Write the recursive form of $Q\pi(s, a)$ and explain the role of recursion. [2 Marks]

Answer Key for Question 5:

(a)

Strategy	Advantage	Drawback
ε-greedy	Simple to implement; ensures exploration	Random exploration may be inefficient
Optimistic Initial Values	Encourages exploration by assuming high rewards	May lead to bias if true rewards are lower
Upper Confidence Bound (UCB)	Balances exploitation and exploration smartly	More complex and computationally demanding

0.5 m for each cell in the above table (advantage/drawback for each method)

(b) $Q_\pi(s,a)=E[R_{t+1}+\gamma \cdot Q_\pi(s_{t+1},a') \mid s_t=s,a_t=a]$

$= \sum_{s',r} p(s',r \mid s,a)(R_{t+1} + \gamma(Q(s_{t+1},a'))$

The recursion captures how the value of taking action a in state s depends not just on the immediate reward R_{t+1}, but also on the value of the next state-action pair under policy π . This backward-looking definition allows Q-values to be updated iteratively during learning, enabling convergence to optimal action values over time.

The **Q-value is defined recursively**: current value = immediate reward + discounted value of the next state-action.

Marking scheme: 1 m for the equation. 1 m for reasoning . value of next state-action pair is the main keyword.

Question -6 [3 +2 = 5 Marks]

(a) You have the following Q-values for four arms of a bandit:

Q1 = 25, Q2 = 32, Q3 = 7, Q4 = 3.

Use ε-greedy with ε = 0.3 to compute the probability of selecting each arm. [2 Marks] Also, identify which arm will be chosen in exploitation. [1 Mark]

(b) What is the difference between stationary and non-stationary multi-armed bandit problems? [1 Mark] Write down one technique which you use to handle non-stationarity. [1 Mark].

Answer Key for Question 6:

(a) With probability (1 − ε) = 0.7, we exploit → choose the best arm (Q2).
With probability ε = 0.3, we explore → choose an arm uniformly at random from all 4 arms.
Each arm has an equal chance (1/4) of being chosen:
→ 0.3 / 4 = 0.075 for each arm
Final Probabilities:
Arm 1: 0.075
Arm 2: 0.775
Arm 3: 0.075
Arm 4: 0.075
Arm chosen in exploitation: Arm 2 (Q2 = 32). Because Q2 has the maximum value.

Marking scheme: 0.5 m for the probability of each arm. (0.5 X 4=2) . 1 m for stating Q2 is chosen at exploitation because it is the maximum value.

(b) In a stationary multi-armed bandit problem, the probability distribution of rewards for each arm remains *constant over time*. In contrast, a non-stationary bandit problem involves changing reward distributions—what worked well before might not work now.
One technique to handle non-stationarity: Discounted Rewards/ exponential recency weighted rewards—these methods focus more on *recent rewards* and gradually forget older data, helping the algorithm adapt to changing environments.

Marking scheme: 0.5m for stationary MAB. 0.5 m for non-stationary MAB. 1m for mentioning the technique to handle with reasoning, as in how it works. Award only 0.5 m if only the method name is mentioned.