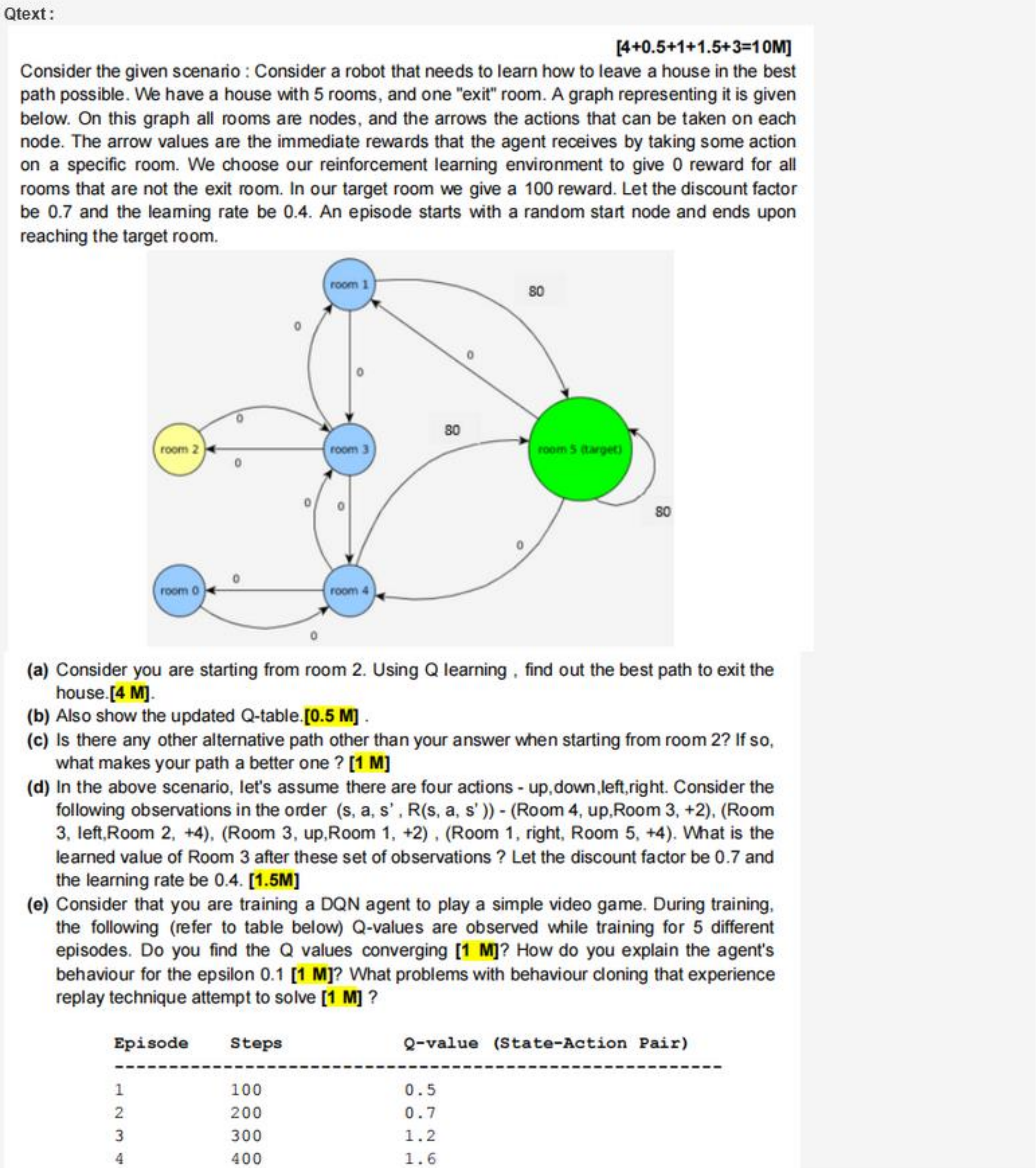
DRL- Previous year October-2023 paper



Qtext :

[4+0.5+1+1.5+3=10M]

Consider the given scenario : Consider a robot that needs to learn how to leave a house in the best path possible. We have a house with 5 rooms, and one "exit" room. A graph representing it is given below. On this graph all rooms are nodes, and the arrows the actions that can be taken on each node. The arrow values are the immediate rewards that the agent receives by taking some action on a specific room. We choose our reinforcement learning environment to give 0 reward for all rooms that are not the exit room. In our target room we give a 100 reward. Let the discount factor be 0.7 and the learning rate be 0.4. An episode starts with a random start node and ends upon reaching the target room.

(a) Consider you are starting from room 2. Using Q learning , find out the best path to exit the house.**[4 M]**.

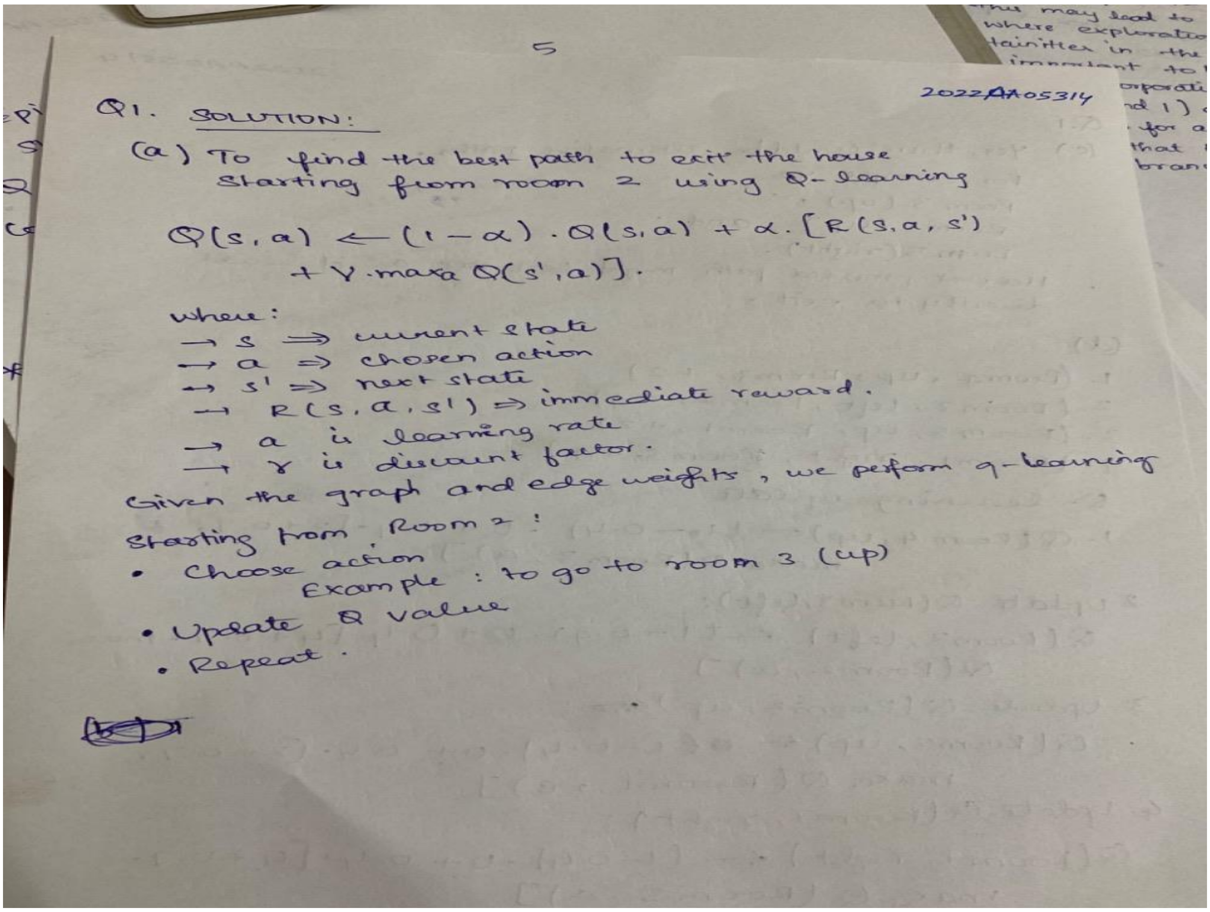
(b) Also show the updated Q-table.**[0.5 M]** .

(c) Is there any other alternative path other than your answer when starting from room 2? If so, what makes your path a better one ? **[1 M]**

(d) In the above scenario, let's assume there are four actions - up,down,left,right. Consider the following observations in the order (s, a, s' , R(s, a, s')) - (Room 4, up,Room 3, +2), (Room 3, left,Room 2, +4), (Room 3, up,Room 1, +2) , (Room 1, right, Room 5, +4). What is the learned value of Room 3 after these set of observations ? Let the discount factor be 0.7 and the learning rate be 0.4. **[1.5M]**

(e) Consider that you are training a DQN agent to play a simple video game. During training, the following (refer to table below) Q-values are observed while training for 5 different episodes. Do you find the Q values converging **[1 M]**? How do you explain the agent's behaviour for the epsilon 0.1 **[1 M]**? What problems with behaviour cloning that experience replay technique attempt to solve **[1 M]** ?

Episode	Steps	Q-value (State-Action Pair)
1	100	0.5
2	200	0.7
3	300	1.2
4	400	1.6



5

2022AA05314

Q1. SOLUTION:

(a) To find the best path to exit the house Starting from room 2 using Q-learning

$$Q(s,a) \leftarrow (1-\alpha)\cdot Q(s,a) + \alpha\cdot[R(s,a,s') + \gamma\cdot \max_a Q(s',a)].$$

where:

→ s ⇒ current state

→ a ⇒ chosen action

→ s' ⇒ next state

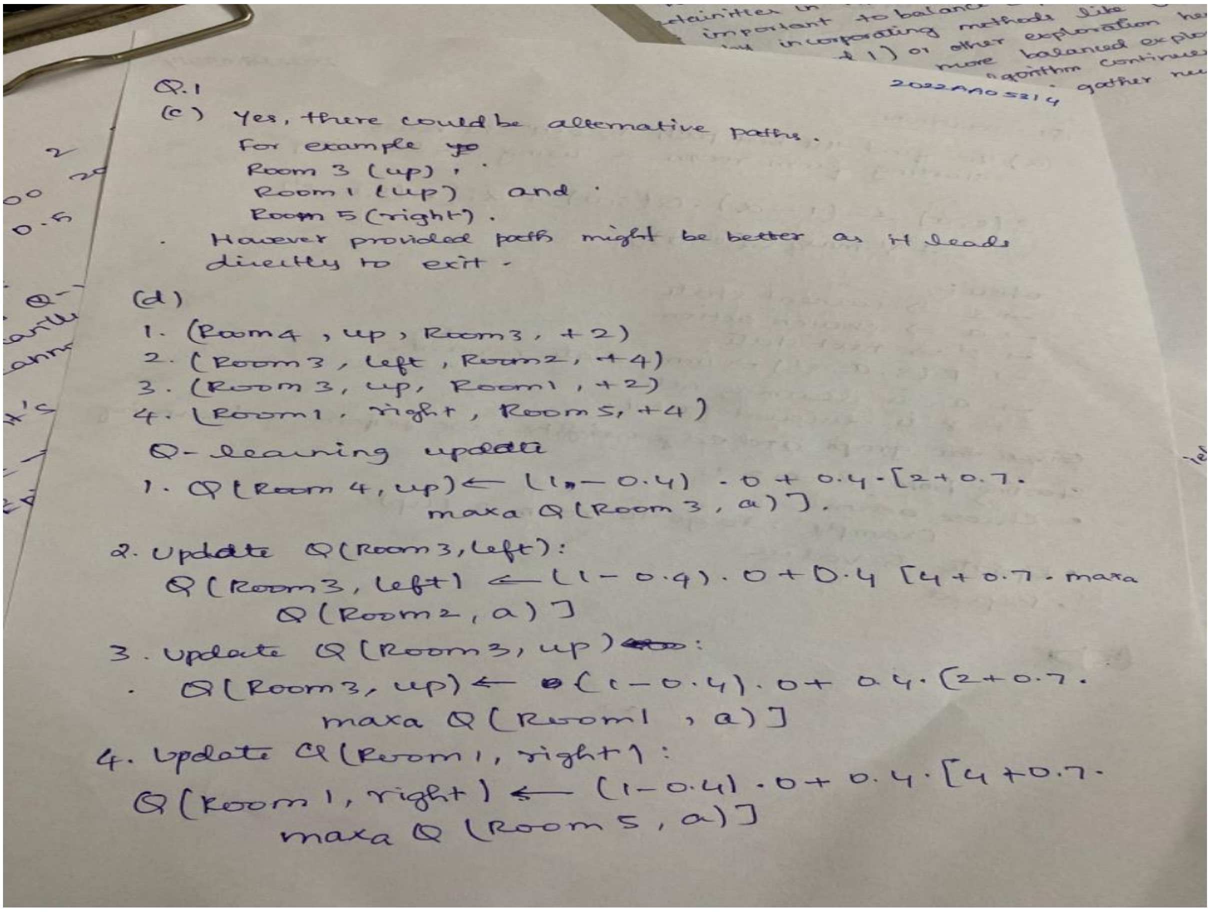
→ R(s, a, s') ⇒ immediate reward.

→ a is learning rate

→ γ is discount factor.

Given the graph and edge weights, we perform q-learning starting from, Room 2!

- Choose action
  Example : to go to room 3 (up)
- Update Q value
- Repeat.



Q.1

2022AA05314

(c) Yes, there could be alternative paths.
For example ~~yo~~
Room 3 (up),
Room 1 (up) and
Room 5 (right).

. However provided path might be better as it leads directly to exit.

(d)

1. (Room4, up, Room3, +2)
2. (Room3, Left, Room2, +4)
3. (Room 3, up, Room1, +2)
4. (Room1, right, Room5, +4)

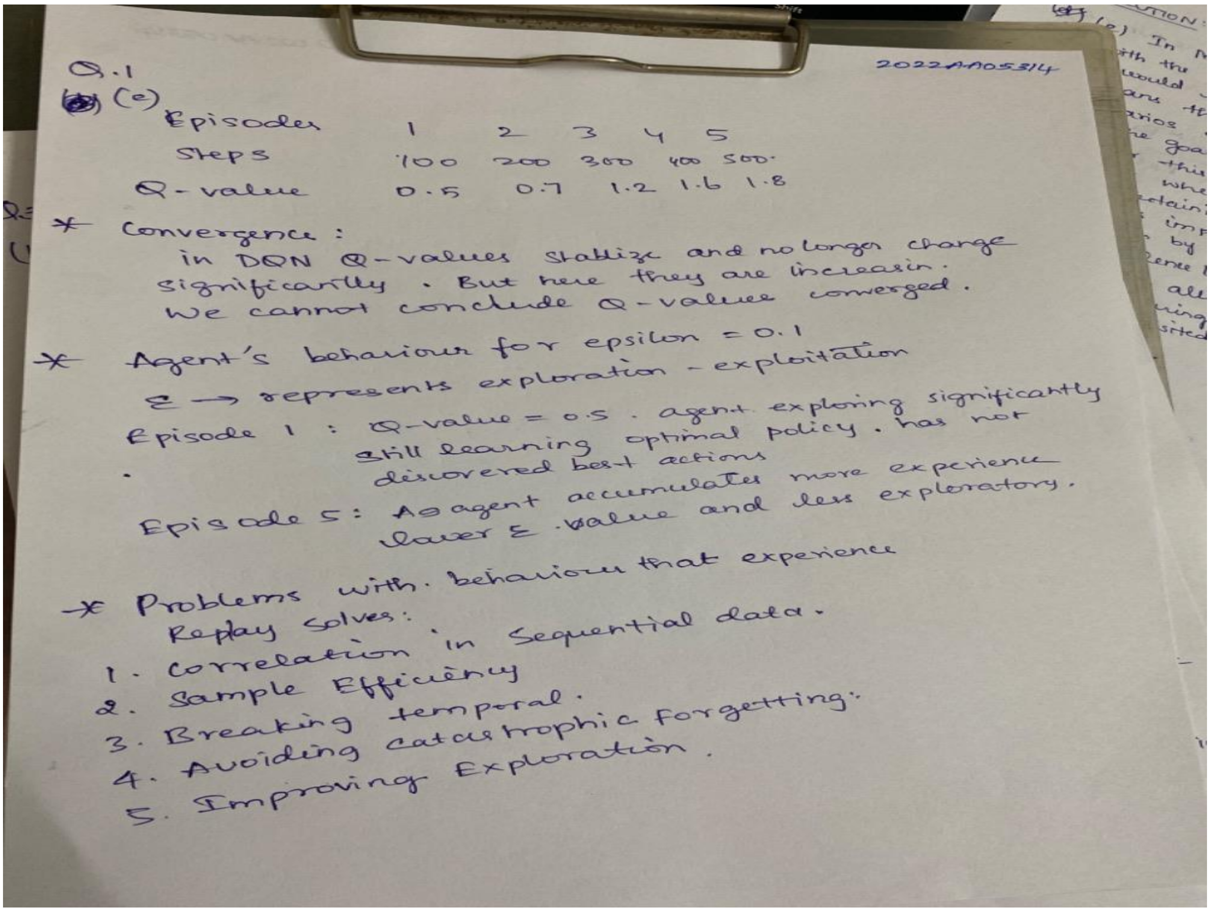
Q-learning update

1. $Q(\text{Room } 4, up) \leftarrow (1-0.4)\cdot 0 + 0.4\cdot[2+0.7\cdot \max_a Q(\text{Room } 3, a)]$.

2. Update Q(Room3, Left):
$Q(\text{Room3}, Left) \leftarrow (1-0.4)\cdot 0 + 0.4\,[4+0.7\cdot \max_a Q(\text{Room2}, a)]$

3. Update Q(Room3, up):
. $Q(\text{Room3}, up) \leftarrow (1-0.4)\cdot 0 + 0.4\cdot[2+0.7\cdot \max_a Q(\text{Room1}, a)]$

4. Update Q(Room1, right):
$Q(\text{Room1}, right) \leftarrow (1-0.4)\cdot 0 + 0.4\cdot[4+0.7\cdot \max_a Q(\text{Room 5}, a)]$



Q.1 2022AA05314

(e)

Episodes	1	2	3	4	5
Steps	100	200	300	400	500.
Q-value	0.5	0.7	1.2	1.6	1.8

* Convergence:
in DQN Q-values stabliz and no longer change significantlly. But here they are increasin.
We cannot conclude Q-values converged.

* Agent's behaviour for epsilon = 0.1
$\varepsilon \rightarrow$ represents exploration - exploitation

Episode 1 : Q-value = 0.5 . agent exploring significantly still learning optimal policy. has not discovered best actions

Episode 5: As agent accumulates more experience lower ε value and less exploratory.

* Problems with behaviour that experience Replay solves:

1. Correlation in Sequential data.
2. Sample Efficiency
3. Breaking temporal.
4. Avoiding catastrophic forgetting.
5. Improving Exploration.

@2-QUESTION:

[3.5+0.5+4+2=10M]

a) Consider the following grid world:

S_0	S_1	S_2	S_3 = 10

The states are S_0, S_1,S_2,S_3 with S_3 being the terminal state with utility value of 10. Move Left and Move right are the two actions possible with a transition probability of 0.2 for left and 0.8 for right respectively. You get a reward of -1 for states and actions except for the terminal which has a utility value of 10. Let the discount factor be 0.25

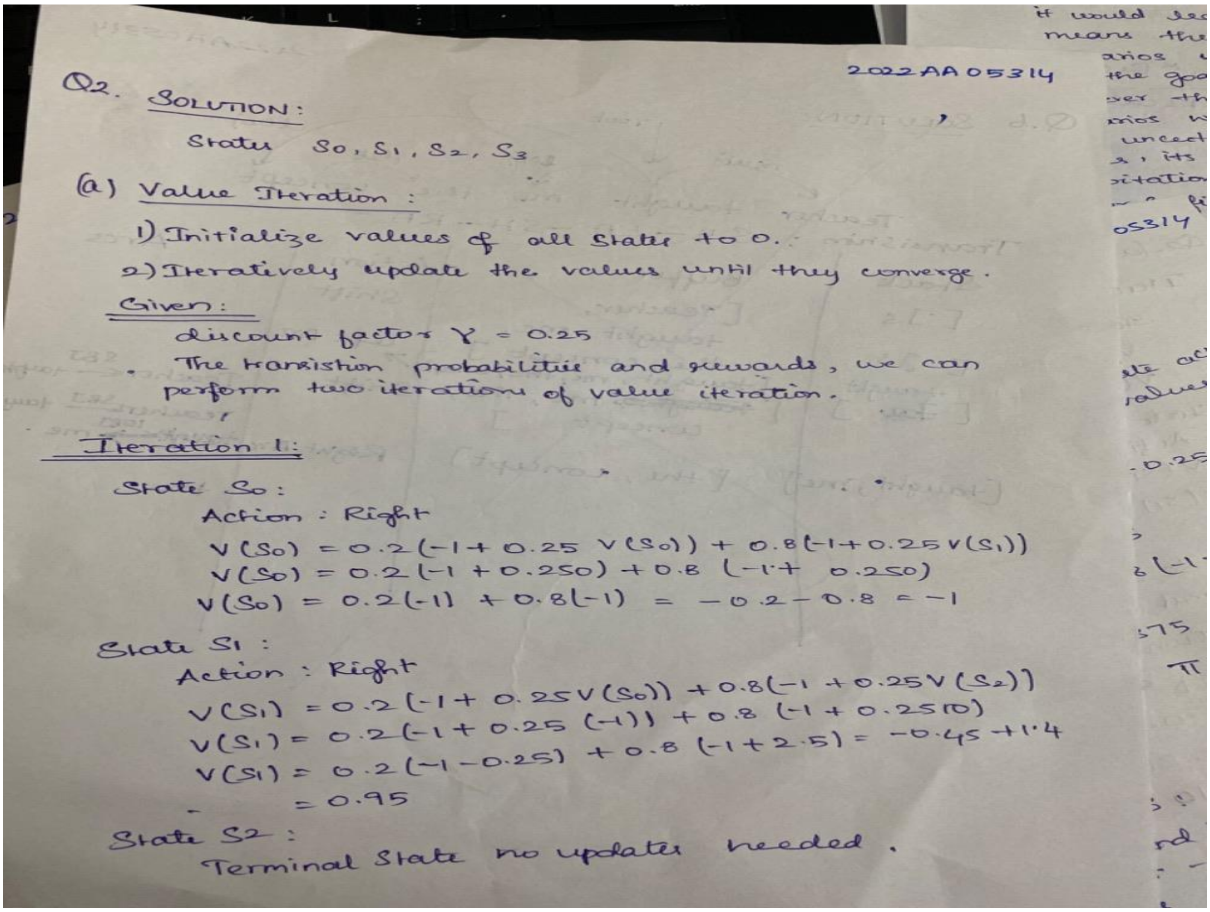
Using value iteration of dynamic programming, determine the values of all states. Let the values of all states be initialised to 0. Show upto 2 Iterations.**[3.5 M]**

(b) What would be the optimal policy as the result of value iteration for this given MDP? **[0.5 M]**

(c) Let b(s|a) = {left(0.46), right(0.54)} and π(s|a) = {right(0.97), left(0.03)}; **τ** = A,up,0; B,up,2; A,down,1; B,up,3; A,up,0; B,down,2

Improve and estimate the policy π such that π≈π*. Use the given b(s|a) and **τ**. Let the initial Q(s, a) for all (s,a) are 0's; and the initial V(s) for all s to be 0's.**[4 M]**

(d) In the given episode **τ** what are the estimates of A and B through every-visit and fist-visit methods? **[2M]**



2022AA05314

Q2. SOLUTION:

States S_0, S_1, S_2, S_3

(a) Value Iteration :

1) Initialize values of all states to 0.

2) Iteratively update the values until they converge.

Given:

discount factor $\gamma = 0.25$

- The transistion probabilities and rewards, we can perform two iterations of value iteration.

Iteration 1:

State S_0:

Action : Right

$V(S_0) = 0.2(-1+0.25\,V(S_0)) + 0.8(-1+0.25\,V(S_1))$

$V(S_0) = 0.2(-1+0.25 0) + 0.8(-1+ 0.25 0)$

$V(S_0) = 0.2(-1) + 0.8(-1) = -0.2 - 0.8 = -1$

State S_1 :

Action : Right

$V(S_1) = 0.2(-1+0.25\,V(S_0)) + 0.8(-1+0.25\,V(S_2))$

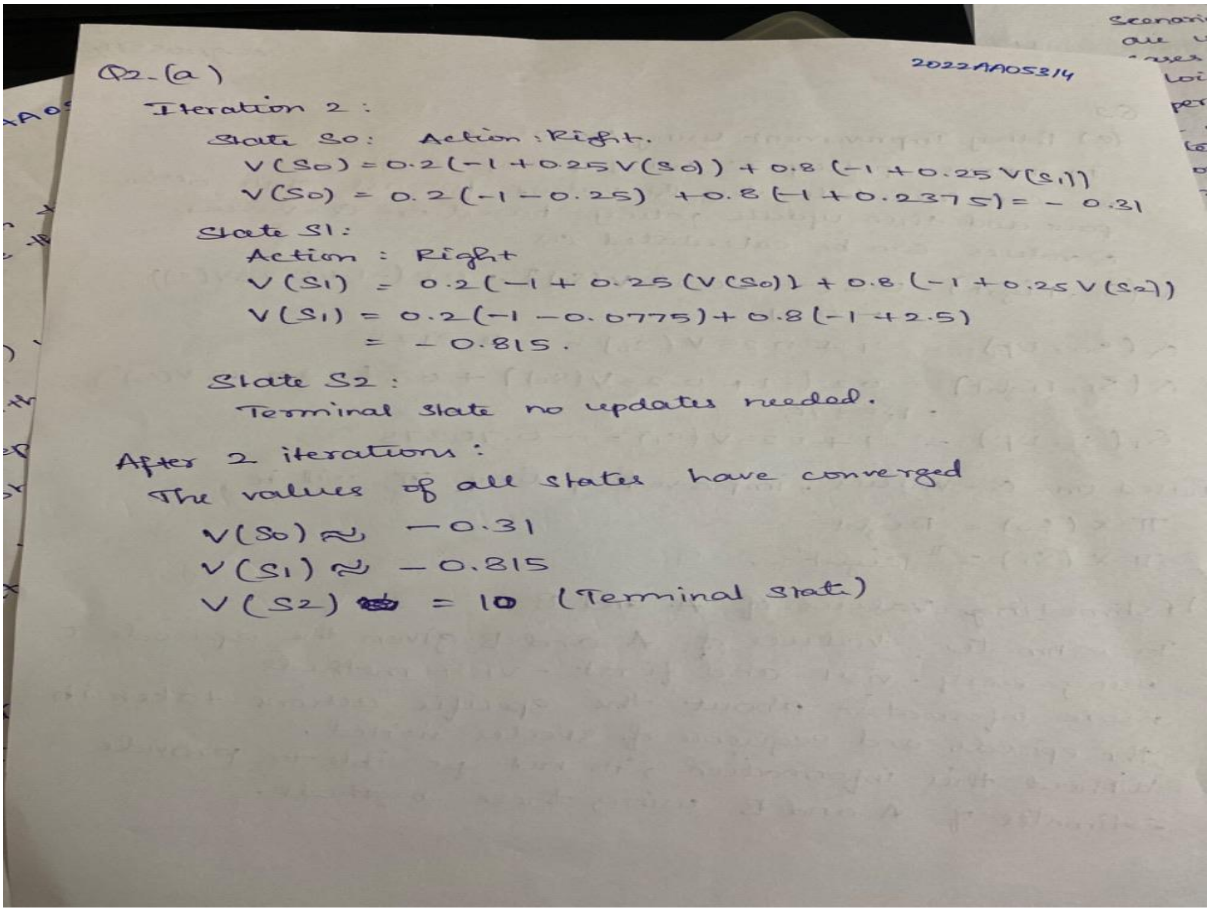
$V(S_1) = 0.2(-1+0.25(-1)) + 0.8(-1+0.25(10)$

$V(S_1) = 0.2(-1-0.25) + 0.8(-1+2.5) = -0.45+1.4$

$= 0.95$

State S_2 :

Terminal State no updates needed.



Q2.(a)

2022AA05314

Iteration 2 :

State S_0: Action : Right.

$V(S_0) = 0.2(-1+0.25\,V(S_0)) + 0.8(-1+0.25\,V(S_1))$

$V(S_0) = 0.2(-1-0.25) + 0.8(-1+0.2375) = -0.31$

State S_1:

Action : Right

$V(S_1) = 0.2(-1+0.25(V(S_0)) + 0.8(-1+0.25\,V(S_2))$

$V(S_1) = 0.2(-1-0.0775) + 0.8(-1+2.5)$

$= -0.815.$

State S_2 :

Terminal state no updates needed.

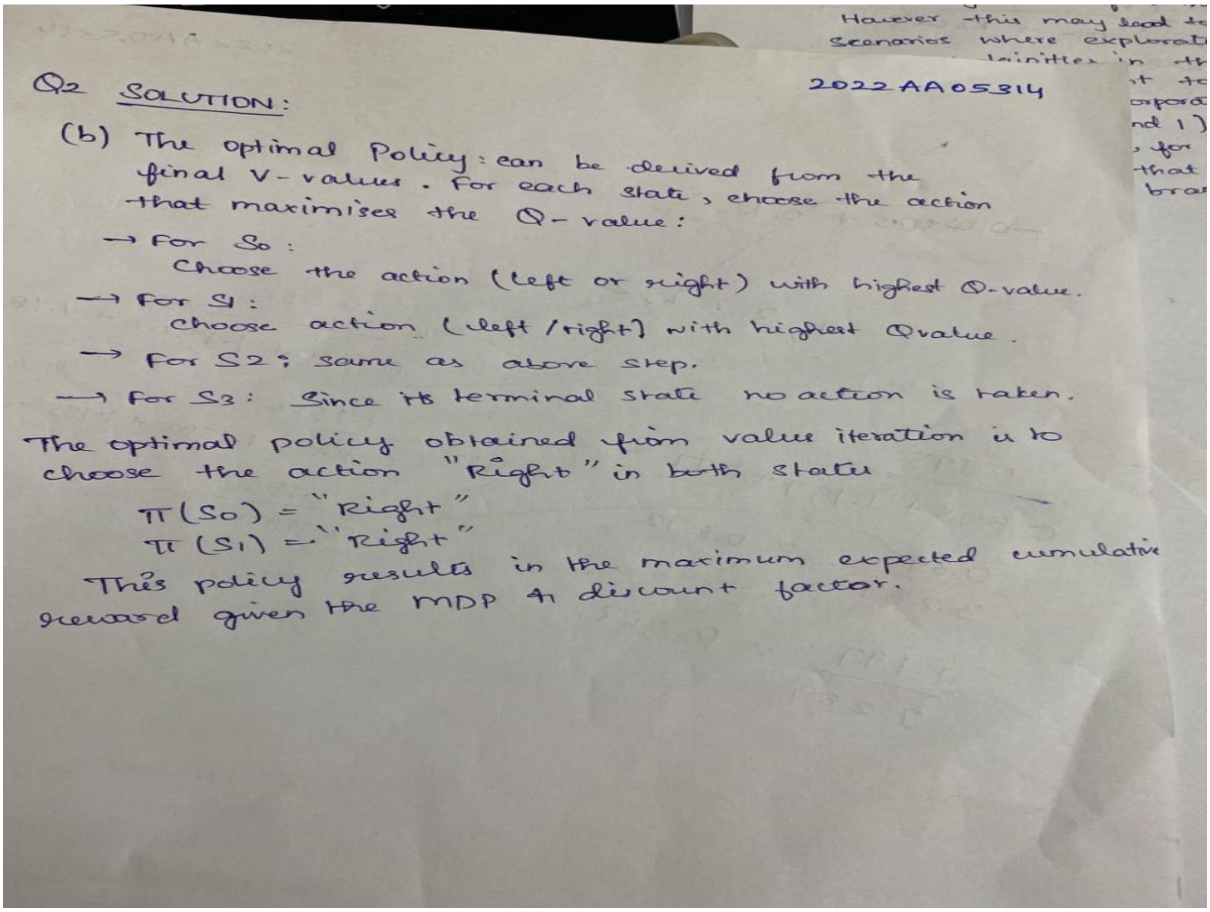
After 2 iterations:

The values of all states have converged

$V(S_0) \approx -0.31$

$V(S_1) \approx -0.815$

$V(S_2) = 10$ (Terminal state)



Q2 SOLUTION:

2022AA05314

(b) The optimal Policy: can be derived from the final V-values. For each state, choose the action that maximises the Q-value:

→ For S_0:
Choose the action (left or right) with highest Q-value.

→ For S_1:
Choose action (left/right) with highest Qvalue.

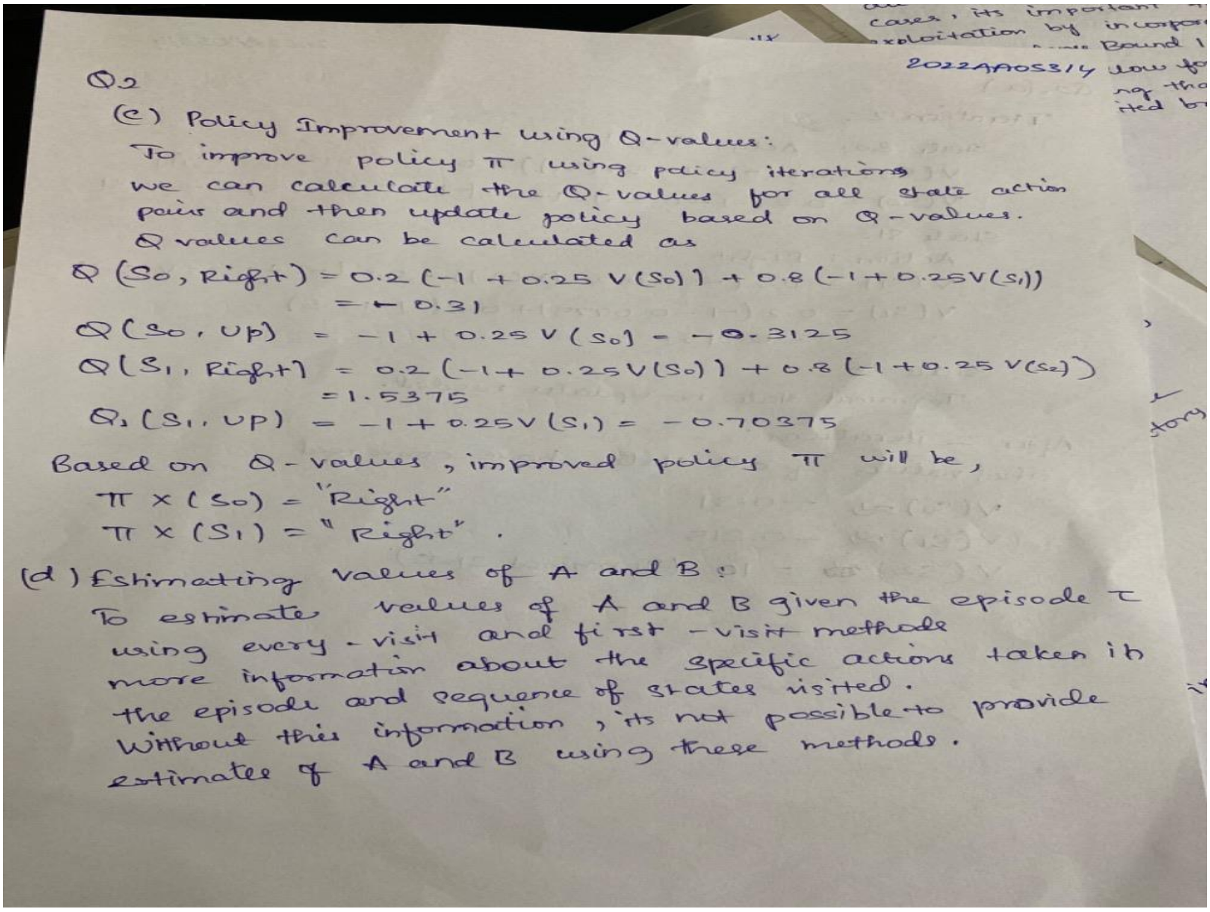
→ For S_2: same as above step.

→ For S_3: Since its terminal state no action is taken.

The optimal policy obtained from value iteration is to choose the action "Right" in both states

$\pi(S_0) =$ "Right"
$\pi(S_1) =$ "Right"

This policy results in the maximum expected cumulative reward given the MDP & discount factor.



Q2

2022AA05314

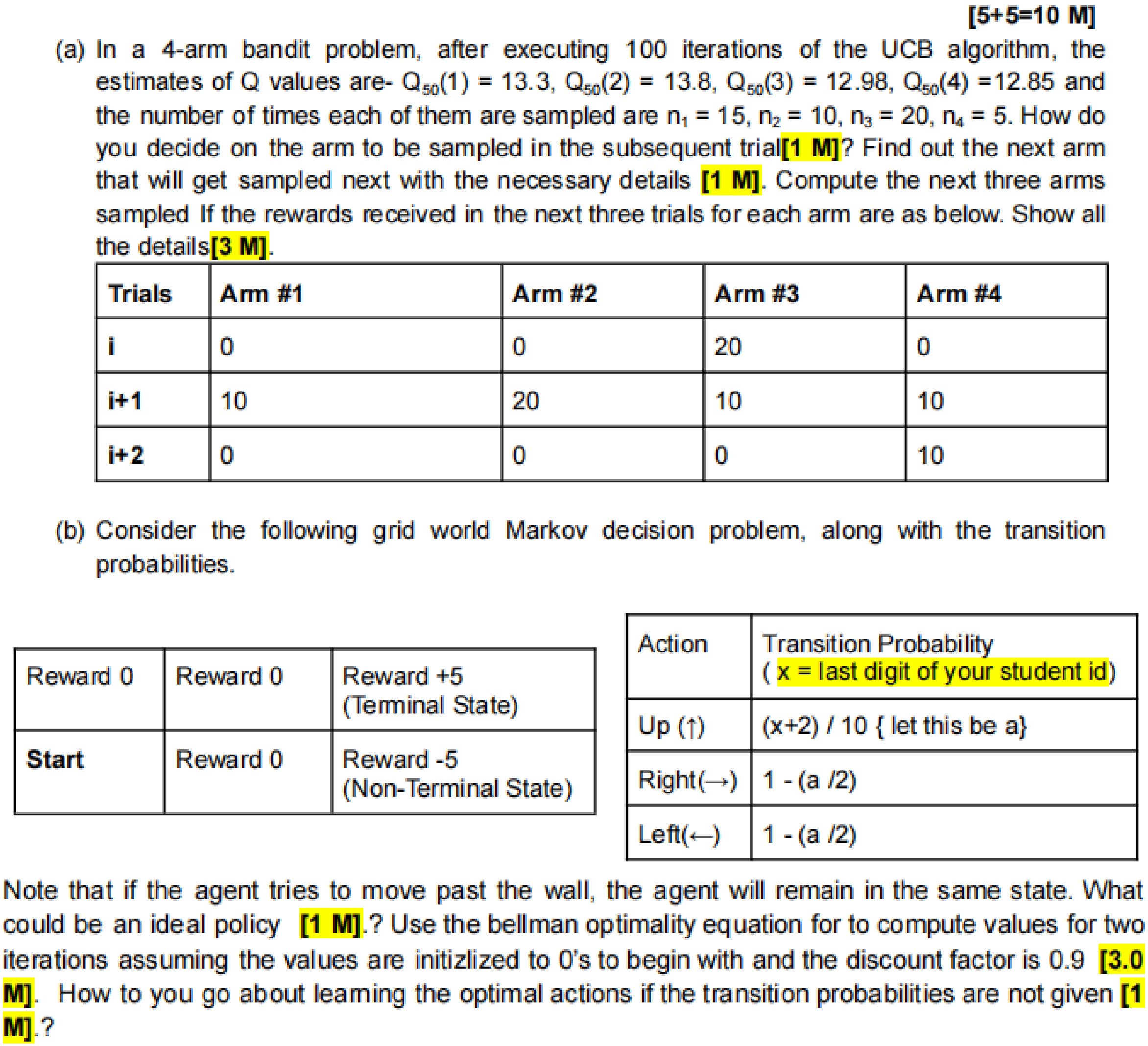
(c) Policy Improvement using Q-values:
To improve policy π using policy iterations we can calculate the Q-values for all state action pairs and then update policy based on Q-values.
Q values can be calculated as

$$Q(S_0, \text{Right}) = 0.2(-1 + 0.25\,V(S_0)) + 0.8(-1 + 0.25V(S_1))$$
$$= -0.31$$
$$Q(S_0, \text{Up}) = -1 + 0.25\,V(S_0) = -0.3125$$
$$Q(S_1, \text{Right}) = 0.2(-1 + 0.25V(S_0)) + 0.8(-1 + 0.25\,V(S_2))$$
$$= 1.5375$$
$$Q_1(S_1, \text{UP}) = -1 + 0.25V(S_1) = -0.70375$$

Based on Q-values, improved policy π will be,
$\pi \times (S_0) =$ "Right"
$\pi \times (S_1) =$ "Right".

(d) Estimating values of A and B:
To estimate values of A and B given the episode τ using every-visit and first-visit methods more information about the specific actions taken in the episode and sequence of states visited.
Without this information, its not possible to provide estimates of A and B using these methods.

#3 QUESTION:



[5+5=10 M]

(a) In a 4-arm bandit problem, after executing 100 iterations of the UCB algorithm, the estimates of Q values are- $Q_{50}(1) = 13.3$, $Q_{50}(2) = 13.8$, $Q_{50}(3) = 12.98$, $Q_{50}(4) = 12.85$ and the number of times each of them are sampled are $n_1 = 15$, $n_2 = 10$, $n_3 = 20$, $n_4 = 5$. How do you decide on the arm to be sampled in the subsequent trial**[1 M]**? Find out the next arm that will get sampled next with the necessary details **[1 M]**. Compute the next three arms sampled If the rewards received in the next three trials for each arm are as below. Show all the details**[3 M]**.

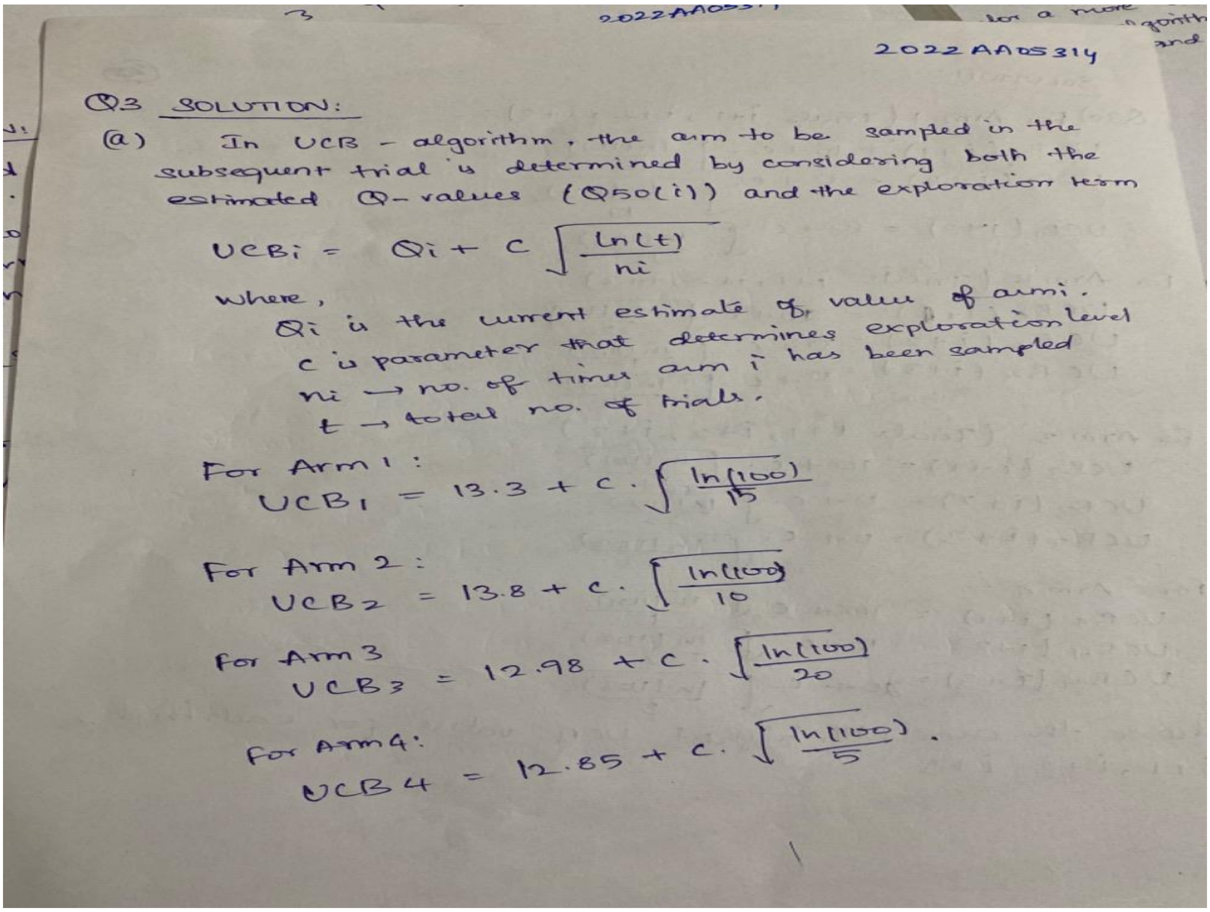
Trials	**Arm #1**	**Arm #2**	**Arm #3**	**Arm #4**
i	0	0	20	0
i+1	10	20	10	10
i+2	0	0	0	10

(b) Consider the following grid world Markov decision problem, along with the transition probabilities.

Reward 0	Reward 0	Reward +5 (Terminal State)
Start	Reward 0	Reward -5 (Non-Terminal State)

Action	Transition Probability (x = last digit of your student id)
Up (↑)	(x+2) / 10 { let this be a}
Right(→)	1 - (a /2)
Left(←)	1 - (a /2)

Note that if the agent tries to move past the wall, the agent will remain in the same state. What could be an ideal policy **[1 M]**.? Use the bellman optimality equation for to compute values for two iterations assuming the values are initizlized to 0's to begin with and the discount factor is 0.9 **[3.0 M]**. How to you go about learning the optimal actions if the transition probabilities are not given **[1 M]**.?



2022 AAD5314

Q3 SOLUTION:

(a) In UCB - algorithm, the arm to be sampled in the subsequent trial is determined by considering both the estimated Q-values (Q50(i)) and the exploration term

$$UCB_i = Q_i + c\sqrt{\frac{\ln(t)}{n_i}}$$

where,

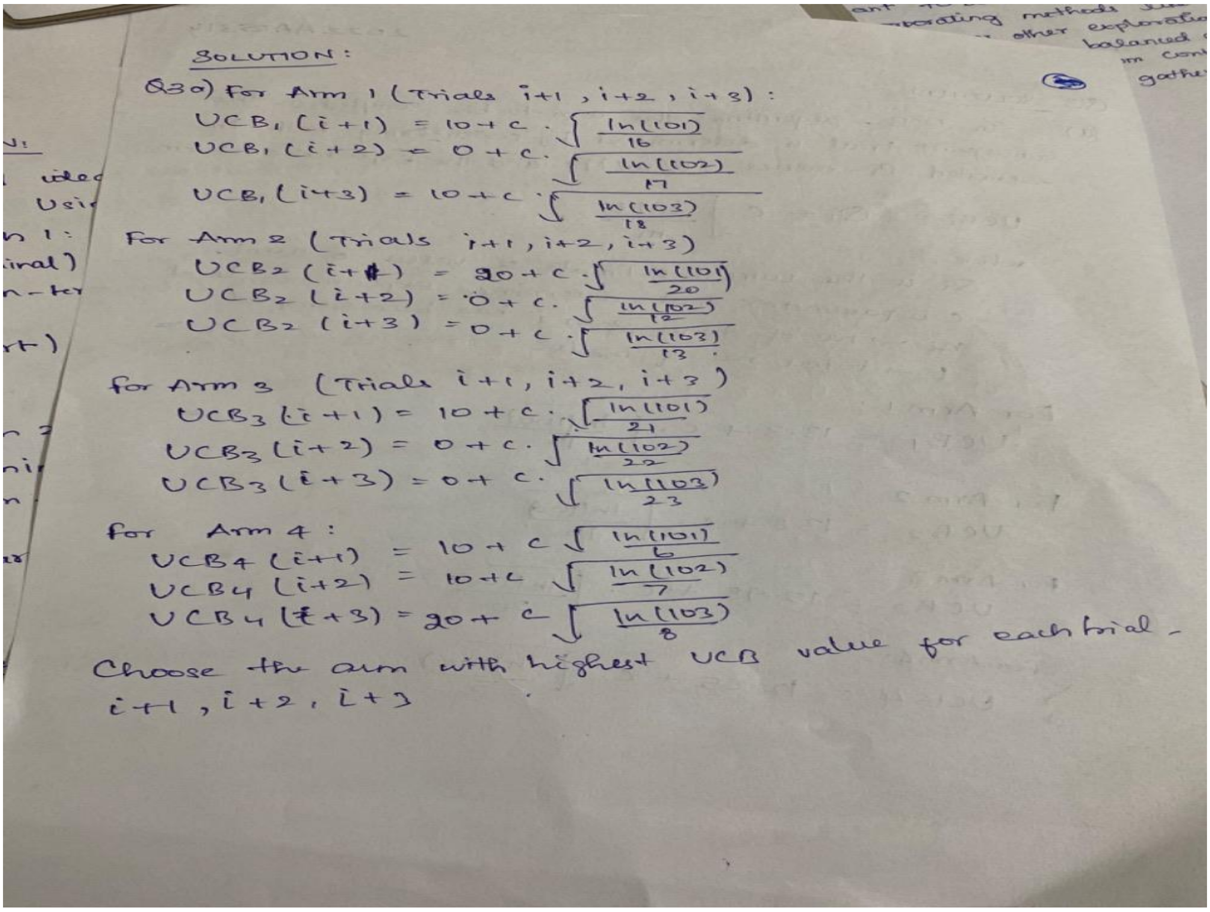
Q_i is the current estimate of value of armi.
c is parameter that determines exploration level
n_i → no. of times arm i has been sampled
t → total no. of trials.

For Arm 1:
$$UCB_1 = 13.3 + c \cdot \sqrt{\frac{\ln(100)}{15}}$$

For Arm 2:
$$UCB_2 = 13.8 + c \cdot \sqrt{\frac{\ln(100)}{10}}$$

For Arm 3
$$UCB_3 = 12.98 + c \cdot \sqrt{\frac{\ln(100)}{20}}$$

For Arm 4:
$$UCB_4 = 12.85 + c \cdot \sqrt{\frac{\ln(100)}{5}}.$$



SOLUTION:

Q3a) For Arm 1 (Trials i+1, i+2, i+3):

$$UCB_1(i+1) = 10 + c \cdot \sqrt{\frac{\ln(101)}{16}}$$
$$UCB_1(i+2) = 0 + c \cdot \sqrt{\frac{\ln(102)}{17}}$$
$$UCB_1(i+3) = 10 + c \cdot \sqrt{\frac{\ln(103)}{18}}$$

For Arm 2 (Trials i+1, i+2, i+3)

$$UCB_2(i+1) = 20 + c \cdot \sqrt{\frac{\ln(101)}{20}}$$
$$UCB_2(i+2) = 0 + c \cdot \sqrt{\frac{\ln(102)}{12}}$$
$$UCB_2(i+3) = 0 + c \cdot \sqrt{\frac{\ln(103)}{13}}.$$

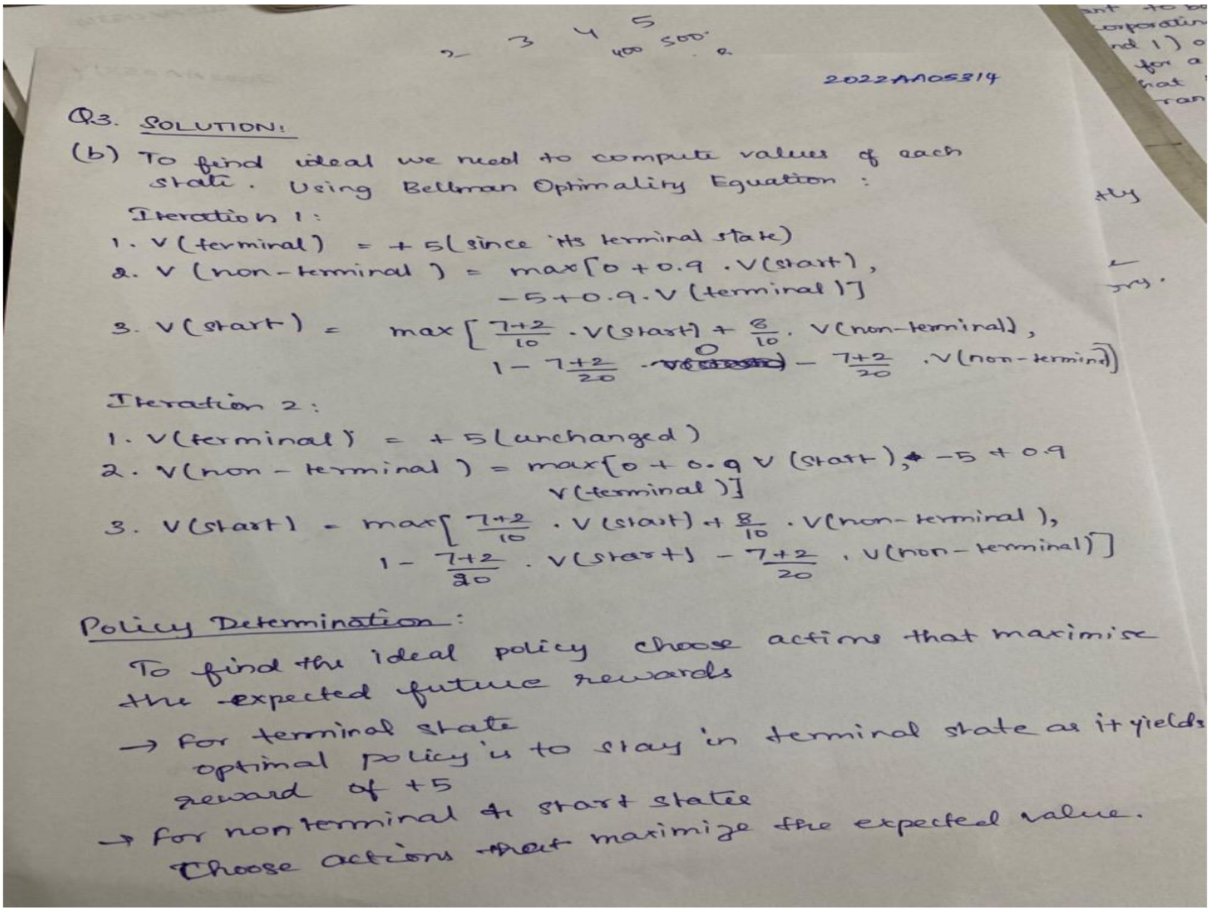
for Arm 3 (Trials i+1, i+2, i+3)

$$UCB_3(i+1) = 10 + c \cdot \sqrt{\frac{\ln(101)}{21}}$$
$$UCB_3(i+2) = 0 + c \cdot \sqrt{\frac{\ln(102)}{22}}$$
$$UCB_3(i+3) = 0 + c \cdot \sqrt{\frac{\ln(103)}{23}}$$

for Arm 4:

$$UCB_4(i+1) = 10 + c\sqrt{\frac{\ln(101)}{6}}$$
$$UCB_4(i+2) = 10 + c\sqrt{\frac{\ln(102)}{7}}$$
$$UCB_4(i+3) = 20 + c\sqrt{\frac{\ln(103)}{8}}$$

Choose the arm with highest UCB value for each trial - i+1, i+2, i+3



2 3 4 5 400 500

2022AA05314

Q3. SOLUTION:

(b) To find ideal we need to compute values of each state. Using Bellman Optimality Equation :

Iteration 1 :

1. $V(\text{terminal}) = +5$ (since its terminal state)
2. $V(\text{non-terminal}) = \max[0 + 0.9 \cdot V(\text{start}), -5 + 0.9 \cdot V(\text{terminal})]$
3. $V(\text{start}) = \max\left[\frac{7+2}{10} \cdot V(\text{start}) + \frac{8}{10} \cdot V(\text{non-terminal}), 1 - 7\frac{+2}{20} \cdot \overset{0}{\cancel{V(\text{start})}} - \frac{7+2}{20} \cdot V(\text{non-terminal})\right]$

Iteration 2 :

1. $V(\text{terminal}) = +5$ (unchanged)
2. $V(\text{non-terminal}) = \max[0 + 0.9\, V(\text{start}), -5 + 0.9\, V(\text{terminal})]$
3. $V(\text{start}) = \max\left[\frac{7+2}{10} \cdot V(\text{start}) + \frac{8}{10} \cdot V(\text{non-terminal}), 1 - \frac{7+2}{20} \cdot V(\text{start}) - \frac{7+2}{20} \cdot V(\text{non-terminal})\right]$

Policy Determination :

To find the ideal policy choose actions that maximise the expected future rewards

→ For terminal state optimal policy is to stay in terminal state as it yields reward of +5

→ For non terminal & start states Choose actions ~~that~~ maximize the expected value.

$4Question:

Answer the following questions: **[2+1.5+2+3+1.5=10M]**

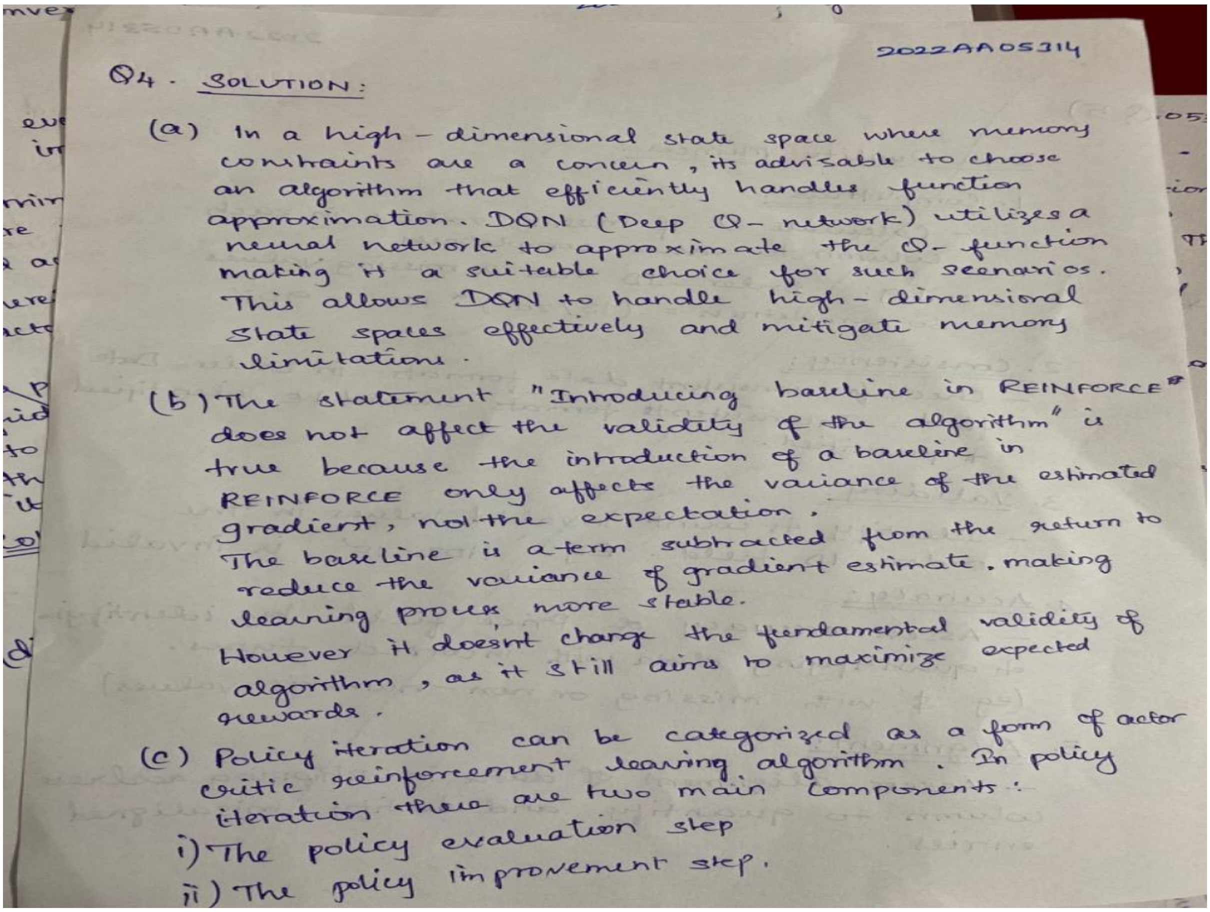
(a) Your state space is high-dimensional and you will run out of memory during computation. You are in need of an algorithm that will definitely reach a local optimum. Would you choose DQN or REINFORCE ? Justify your answer. **[2M]**

(b) Reason for the statement - " Introducing baseline in REINFORCE does not affect the validity of the algorithm ". **[1.5M]**

(c) Can you categorise policy iteration as an actor-critic reinforcement algorithm ? Justify your answer. **[2 M]**

(d) Consider the grid world MDP given in the question paper or use any version of grid world MDP that you are comfortable with. Provide a formulation of this problem in a way that we can perform imitation learning using DAGGER and demonstrate how DAGGER solves this problem using 2 iterations. **[3 M]**

(e) How is exploration and exploitation tradeoff balanced in Monte Carlo Tree search if action with best Q value is chosen at each branch? **[1.5 M]**



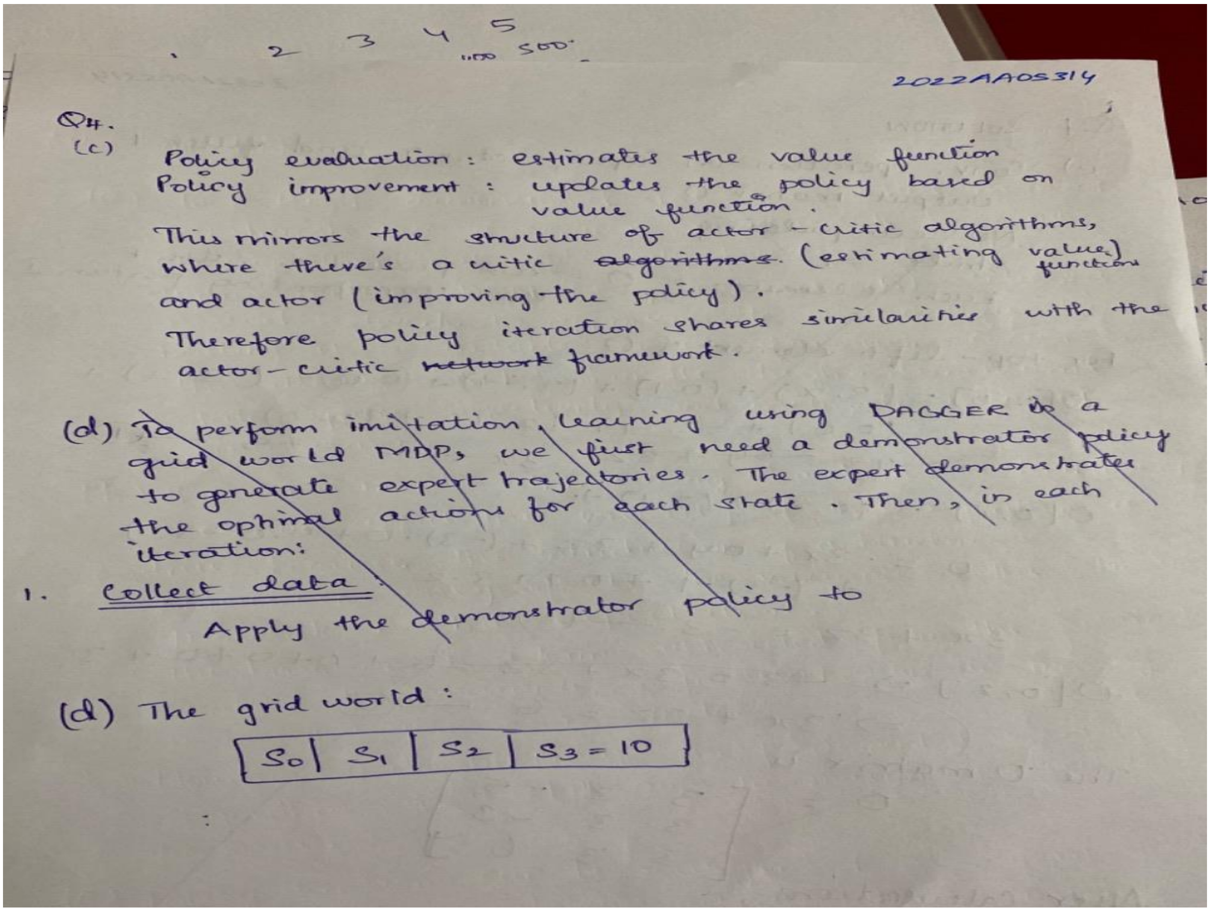
2022AA05314

Q4. SOLUTION:

(a) In a high-dimensional state space where memory constraints are a concern, its advisable to choose an algorithm that efficiently handles function approximation. DQN (Deep Q-network) utilizes a neural network to approximate the Q-function making it a suitable choice for such scenarios. This allows DQN to handle high-dimensional State spaces effectively and mitigate memory limitations.

(b) The statement "Introducing baseline in REINFORCE" does not affect the validity of the algorithm" is true because the introduction of a baseline in REINFORCE only affects the variance of the estimated gradient, not the expectation.
The baseline is a term subtracted from the return to reduce the variance of gradient estimate, making learning process more stable.
However it doesnt change the fundamental validity of algorithm, as it still aims to maximize expected rewards.

(c) Policy iteration can be categorized as a form of actor critic reinforcement leaving algorithm. In policy iteration there are two main components:
i) The policy evaluation step
ii) The policy improvement step.



2022AA05314

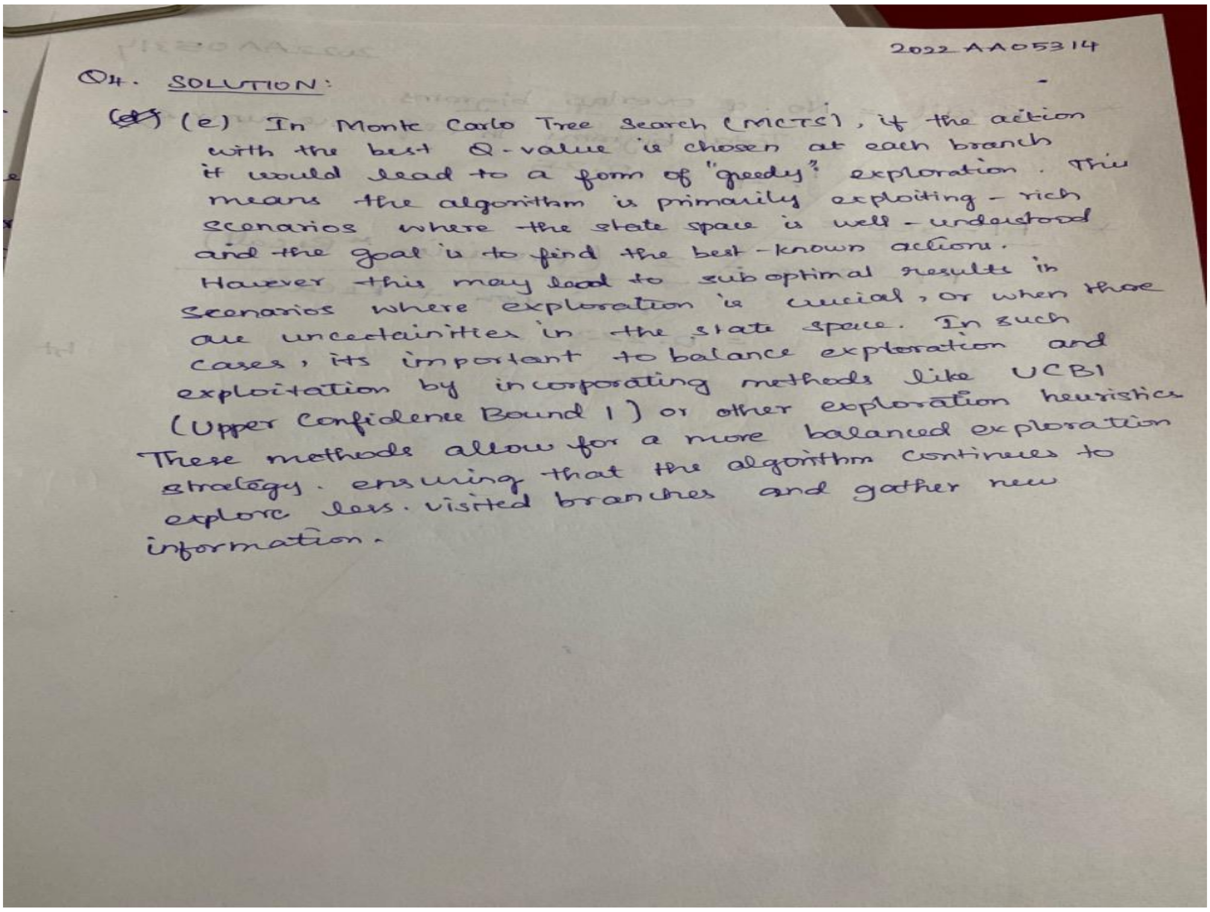
Q4.
(c)
Policy evaluation : estimates the value function
Policy improvement : updates the policy based on value function.
This mirrors the structure of actor-critic algorithms, where there's a critic ~~algorithms~~. (estimating value function) and actor (improving the policy).
Therefore policy iteration shares similarities with the actor-critic ~~network~~ framework.

~~(d) To perform imitation learning using DAGGER in a grid world MDP, we first need a demonstrator policy to generate expert trajectories. The expert demonstrates the optimal actions for each state. Then, in each iteration:~~

~~1. Collect data~~
~~Apply the demonstrator policy to~~

(d) The grid world :

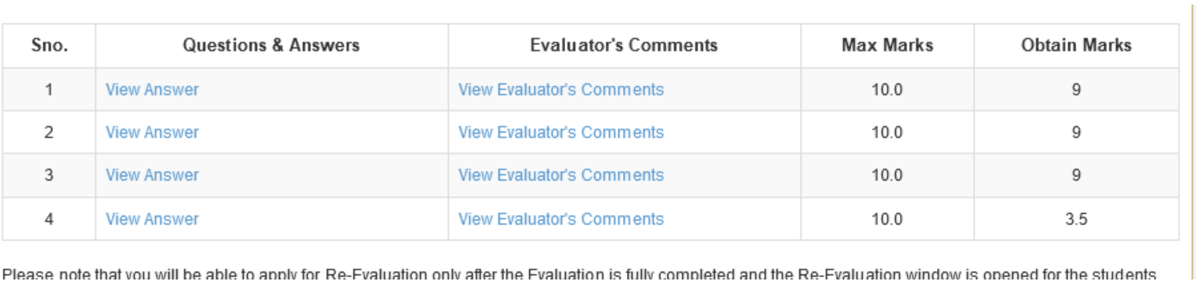
S_0	S_1	S_2	$S_3 = 10$



2022AA05314

Q4. SOLUTION:

(e) In Monte Carlo Tree Search (MCTS), if the action with the best Q-value is chosen at each branch it would lead to a form of "greedy" exploration. This means the algorithm is primarily exploiting - rich scenarios where the state space is well-understood and the goal is to find the best-known actions. However this may lead to suboptimal results in scenarios where exploration is crucial, or when those are uncertainities in the state space. In such cases, its important to balance exploration and exploitation by incorporating methods like UCB1 (Upper Confidence Bound 1) or other exploration heuristics These methods allow for a more balanced exploration strategy. ensuring that the algorithm continues to explore less. visited branches and gather new information.



Sno.	Questions & Answers	Evaluator's Comments	Max Marks	Obtain Marks
1	View Answer	View Evaluator's Comments	10.0	9
2	View Answer	View Evaluator's Comments	10.0	9
3	View Answer	View Evaluator's Comments	10.0	9
4	View Answer	View Evaluator's Comments	10.0	3.5

Please note that you will be able to apply for Re-Evaluation only after the Evaluation is fully completed and the Re-Evaluation window is opened for the students