2) Which of the following choices is **NOT** a component of MDP. [3 points]
Learning rate

3) What are the differences between model-free and model-based reinforcement learning? [5 points]
In model-based reinforcement learning we learn a model of the environment that is the learn transition matrix and discover the rewards. But in model-free we only learn the values of states and actions.

4) Consider a robot movement in the 2D environment given below. Assume that the robot can only take right (R) or left (L) actions. The state-space contains six distinct states {S1, S2, S3, S4}. S1 and S4 are the terminal states with -1 and +2 rewards respectively. Assume that state transitions are stochastic. When trying to move in a certain direction, the robot succeeds with a probability of 0.85. With a probability of 0.1, it remains in the same state, and it may even move in the opposite direction with a probability of 0.05. [8 points]

r= -1 r= +2

S1	S2	S3	S4

Consider the Q-values for all state-action pairs given in the table below:

Q(State, Action)	L	R
S1	0	0
S2	-0.8	0.6
S3	-0.1	2
S4	0	0

If the discount factor is 0.9:

A. Apply Q-learning for one iteration and update the Q-values for all state-action pairs. [6 points]

Q_{k+1} (s,a) = E[R(s,a,s') + discount factor * max(Q_{k}(s',a'))

Q_{1}(s1,**L**) =

0.85*[R(s1,L,s1) + 0.9 * max(Q_{0}(s1,a'))] +

0.1*[R(s1,L,s1) + 0.9* max(Q_{0}(s1,a'))] +

0.05*[R(s1,R,s1) + 0.9 * max(Q_{0}(s1,a'))] =

0.85*[-1 + 0.9*0] + 0.1*[-1 + 0.9*0] + 0.05*[-1 + 0.9*0] = **-1**

Q_{k+1} (s,a) = E[R(s,a,s') + discount factor * max(Q_{k}(s',a'))

Q_{1}(s1,**R**) =

0.85*[R(s1,R,s1) + 0.9 * max(Q_{0}(s1,a'))] +

0.1*[R(s1,R,s1) + 0.9* max(Q_{0}(s1,a'))] +

0.05*[R(s1,L,s1) + 0.9 * max(Q_{0}(s1,a'))] = 0.85*[-1 + 0.9*0] + 0.1*[-1 + 0.9*0] + 0.05*[-1 + 0.9*0] = **-1**

Q_{1}(s2,**L**) =

0.85*[R(s2,L,s1) + 0.9 * max(Q_{0}(s1,a'))] +

0.1*[R(s2,L,s2) + 0.9* max(Q_{0}(s2,a'))] +

0.05*[R(s2,R,s3) + 0.9 * max(Q_{0}(s3,a'))]

= 0.85*[-1 + 0.9*0] + 0.1*[0 + 0.9*0.6]+0.05 *[0 + 0.9*2] = **-0.706**

Q_{1}(s2,**R**) =

0.85*[R(s2,R,s3) + 0.9 * max(Q_{0}(s3,a'))] +

0.1*[R(s2,R,s2) + 0.9* max(Q_{0}(s2,a'))] +

0.05*[R(s2,L,s1) + 0.9 * max(Q_{0}(s1,a'))]

= 0.85*[0 + 0.9*2] + 0.1*[0 + 0.9*0.6]+0.05 *[-1 + 0.9*0] = **1.534**

Q_{1}(s3,**L**) =

0.85*[R(s3,L,s2) + 0.9 * max(Q_{0}(s2,a'))] +

0.1*[R(s3,L,s3) + 0.9* max(Q_{0}(s3,a'))] +

0.05*[R(s3,R,s4) + 0.9 * max(Q_{0}(s4,a'))]

= 0.85*[0 + 0.9*0.6] + 0.1* [0 + 0.9*2] + 0.05*[2 + 0.9*0] = **0.739**

Q_{1}(s3,**R**) =

0.85*[R(s3,R,s4) + 0.9 * max(Q_{0}(s4,a'))] +

0.1*[R(s3,R,s3) + 0.9* max(Q_{0}(s3,a'))] +

0.05*[R(s3,L,s2) + 0.9 * max(Q_{0}(s2,a'))]

= 0.85 * [2 + 0.9*0] + 0.10 * [0 + 0.9*2] + 0.05*[0 + 0.9*0.6] = **1.907**

Q_{1}(s4,**L**) =

0.85*[R(s4,L,s4) + 0.9 * max(Q_{0}(s4,a'))] +

0.1*[R(s4,L,s4) + 0.9* max(Q_{0}(s4,a'))] +

0.05*[R(s4,R,s4) + 0.9 * max(Q_{0}(s4,a'))]

= 0.85 * [2 + 0.9*0] + 0.1*[2 + 0.9*0] + 0.05*[2 + 0.9*0]

= **2**

Q_{1}(s4,**R**) =

0.85*[R(s4,R,s4) + 0.9 * max(Q_{0}(s4,a'))] +

0.1*[R(s4,L,s4) + 0.9* max(Q_{0}(s4,a'))] +

0.05*[R(s4,R,s4) + 0.9 * max(Q_{0}(s4,a'))]

= 0.85 * [2 + 0.9*0] + 0.1*[2 + 0.9*0] + 0.05*[2 + 0.9*0]

= **2**

Q(State, Action)	**L**	**R**
S1	**-1**	**-1**
S2	**-0.706**	**1.534**
S3	**0.739**	**1.907**
S4	**2**	**2**

B. Based on the Q-values that you found in section A, find a sub-optimal/optimal policy. [2 points]

A policy is found by π(s) = arg max Q(s,a)

Since S1 and S4 are terminal states any action won't change states.

We want π(S2) = R and π(S3) = R. Therefore the optimal policy is to go right when not in a terminal state.

Note: round your calculations to two decimal places.

5) Given the optimum value for states $\mathbf{V^*(s)}$, can you compute the optimum policy

$\boldsymbol{\pi^*(s)}$, if you do not know the transition probabilities, **T**, and the reward function, **R**? Explain why or why not. [4 points]

No, because

$\pi^*(s) = argmax \Sigma T(s,a,s')[R(s,a,s') + \gamma V^*(s')]$, R and T are needed to extract a policy.

6)

A. Describe two methods for creating a balance between exploration and exploitation in active Reinforcement Learning problems. [3 points]

i) Use ε −greedy method to explore with probability ε and act on current policy with 1-ε probability.

ii) Use an exploration function to map value estimates and the number of times a state-action pair has been encountered to some real number, then use this number in addition to the observed reward to update Q-values.

B. What is the purpose of exploration in active RL problems? [3 points]

When the learning rate is high, our policy does not know the best actions yet so we must explore and try new actions and observe the reward. Over time our policy will act with less randomness and have smaller updates to model values.

7)

A. Describe the core formula for TD learning. [3 points]

TD-learning updates V(s) each time we encounter a new transition (s, a, s', r), we denote a sample as, sample = R(s,$\pi(s)$,s') + γV^{π}(s') and we update values by the following, $V^{\pi}(s) \leftarrow (1-\alpha)V^{\pi}(s) + \alpha * sample$

B. What is the role of the learning rate? [3 points]

The learning rate α determines the weighting of the current estimate compared to the old estimate.

C. What is the benefit of Q-learning over TD learning? [3 points]

Q-learning allows the learning of Q-values rather than just V-values, which makes action selection model-free. This allows for finding new policies while in TD-learning the policy is fixed.