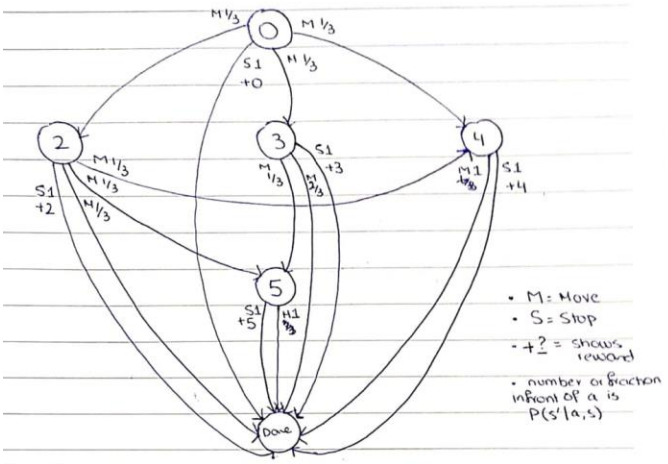
ASSIGNMENT #3

Muhammad Umar Salman
21010241

Markov Decision Processes

Imagine a modification of the racing car as discussed in lecture. In this game, the car repeatedly moves a random number of spaces that is equally likely to be 2, 3, or 4. The car can either Move or Stop if the total number of spaces moved is less than 6.
If the total spaces moved is 6 or higher, the game automatically ends, and the car receives a reward of 0. When the car Stops, the reward is equal to the total spaces moved (up to 5), and the game ends. There is no reward for the Move action.
Let's formulate this problem as an MDP with the states {0, 2, 3, 4, 5, Done}.

(a) What is the transition function for this MDP? (Specify discrete values for specific state/action inputs.)



Ans a)

$T(0, M, s') = 1/3$ where $s' \in \{2, 3, 4\}$
$T(2, M, s') = 1/3$ where $s' \in \{4, 5, Done\}$
$T(3, M, 5) = 1/3$
$T(3, M, Done) = 2/3$
$T(4, M, Done) = 1$
$T(5, M, Done) = 1$
$T(s, S, Done) = 1$ where $s \in \{0, 2, 3, 4, 5\}$
$T(s, a, s') = 0$ For everything else

(b) What is the reward function for this MDP?

Ans b) $R(0, S, \text{Done}) = 0$
$R(2, S, \text{Done}) = 2$
$R(3, S, \text{Done}) = 3$
$R(4, S, \text{Done}) = 4$
$R(5, S, \text{Done}) = 5$
$R(s, a, s') = 0$ for everything else.

(c) Use the value iteration update equation and perform value iteration updates for four iterations with γ = 1.0

States	0	2	3	4	5
V_0	0	0	0	0	0
V_1	0	2	3	4	5
V_2	3	3	3	4	5
V_3	10/3	3	3	4	5
V_4	10/3	3	3	4	5

V1 Calculations

① Initialize $V_0 = \vec{0}$

$V_1(0) = \max_a \begin{cases} Q_1(0,M) = \frac{1}{3}(0+1(0)) + \frac{1}{3}(0+1(0)) + \frac{1}{3}(0+1(0)) = 0 \\ Q_1(0,S) = 1(0+1(0)) = 0 \end{cases}$
$\max(Q_1(0,M), Q_1(0,S)) = 0$

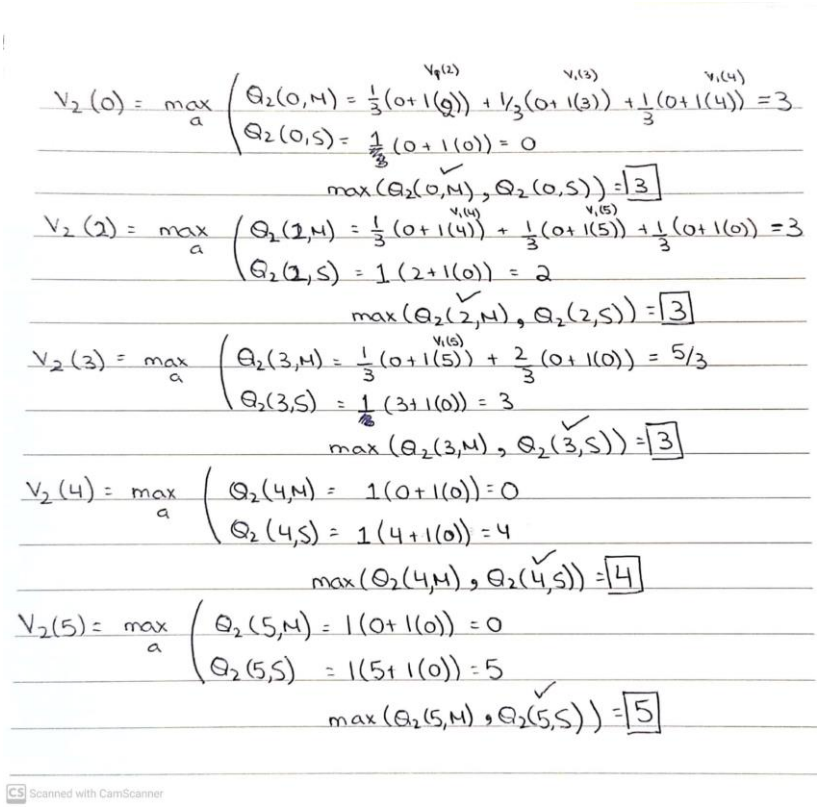
$V_1(2) = \max_a \begin{cases} Q_1(2,M) = \frac{1}{3}(0+1(0)) + \frac{1}{3}(0+1(0)) + \frac{1}{3}(0+1(0)) = 0 \\ Q_1(2,S) = 1(2+1(0) = 2 \end{cases}$
$\max(Q_1(2,M), Q_1(0,S)) = 2$

$V_1(3) = \max_a \begin{cases} Q_1(3,M) = \frac{1}{3}(0+1(0)) + \frac{2}{3}(0+1(0)) = 0 \\ Q_1(3,S) = 1(3+1(0)) = 3 \end{cases}$
$\max(Q_1(3,M), Q_1(3,S)) = 3$

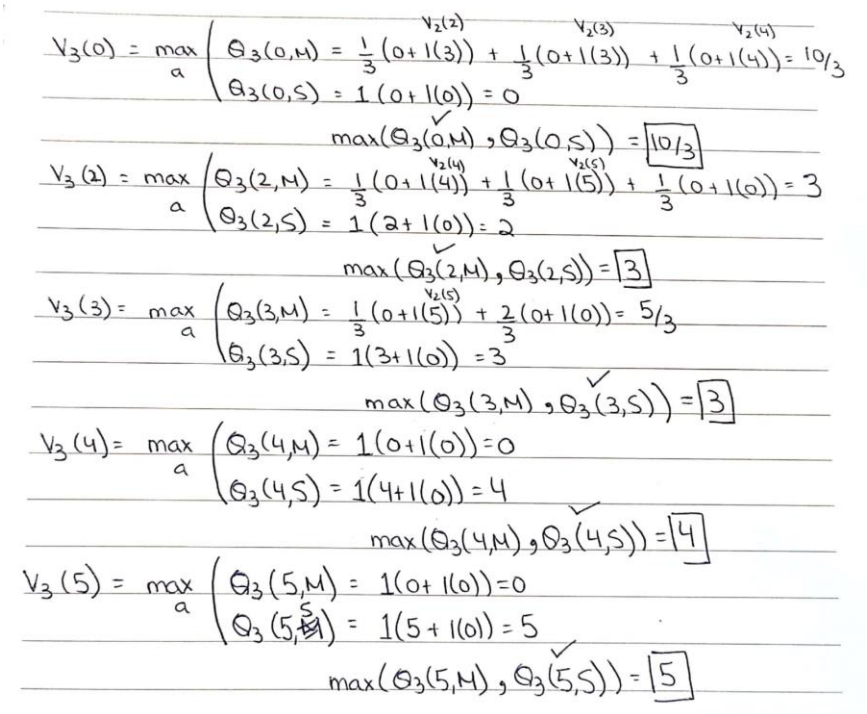
$V_1(4) = \max_a \begin{cases} Q_1(4,M) = 1(0+1(0)) = 0 \\ Q_1(4,S) = 1(4+1(0)) = 4 \end{cases}$
$\max(Q_1(4,M), Q_1(4,S)) = 4$

$V_1(5) = \max_a \begin{cases} Q_1(5,M) = 1(0+1(0)) = 0 \\ Q_1(5,S) = 1(5+1(0)) = 5 \end{cases}$
$\max(Q_1(5,M), Q_1(5,S)) = 5$

V2 Calculations



V3 Calculations



V4 Calculations

$V_4(0) = \max_a \begin{cases} Q_4(0,M) = \frac{1}{3}(0+1(V_3(2))) + \frac{1}{3}(0+1(V_3(3))) + \frac{1}{3}(0+1(V_3(4))) = 10/3 \\ Q_4(0,S) = 1(0+1(0)) = 0 \end{cases}$

$\max(Q_4(0,M), Q_4(0,S)) = \boxed{10/3}$

$V_4(2) = \max_a \begin{cases} Q_4(2,M) = \frac{1}{3}(0+1(V_3(4))) + \frac{1}{3}(0+1(V_3(5))) + \frac{1}{3}(0+1(0)) = 3 \\ Q_4(2,S) = 1(2+1(0)) = 2 \end{cases}$

$\max(Q_4(2,M), Q_4(2,S)) = \boxed{3}$

$V_4(3) = \max_a \begin{cases} Q_4(3,M) = \frac{1}{3}(0+1(V_3(5))) + \frac{1}{3}(0+1(0)) = 5/3 \\ Q_4(3,S) = 1(3+1(0)) = 3 \end{cases}$

$\max(Q_4(3,M), Q_4(3,S)) = \boxed{3}$

$V_4(4) = \max_a \begin{cases} Q_4(4,M) = 1(0+1(0)) = 0 \\ Q_4(4,S) = 1(4+1(0)) = 4 \end{cases}$

$\max(Q_4(4,M), Q_4(4,S)) = \boxed{4}$

$V_4(5) = \max_a \begin{cases} Q_4(5,M) = 1(0+1(0)) = 0 \\ Q_4(5,S) = 1(5+1(0)) = 5 \end{cases}$

$\max(Q_4(5,M), Q_4(5,S)) = \boxed{5}$

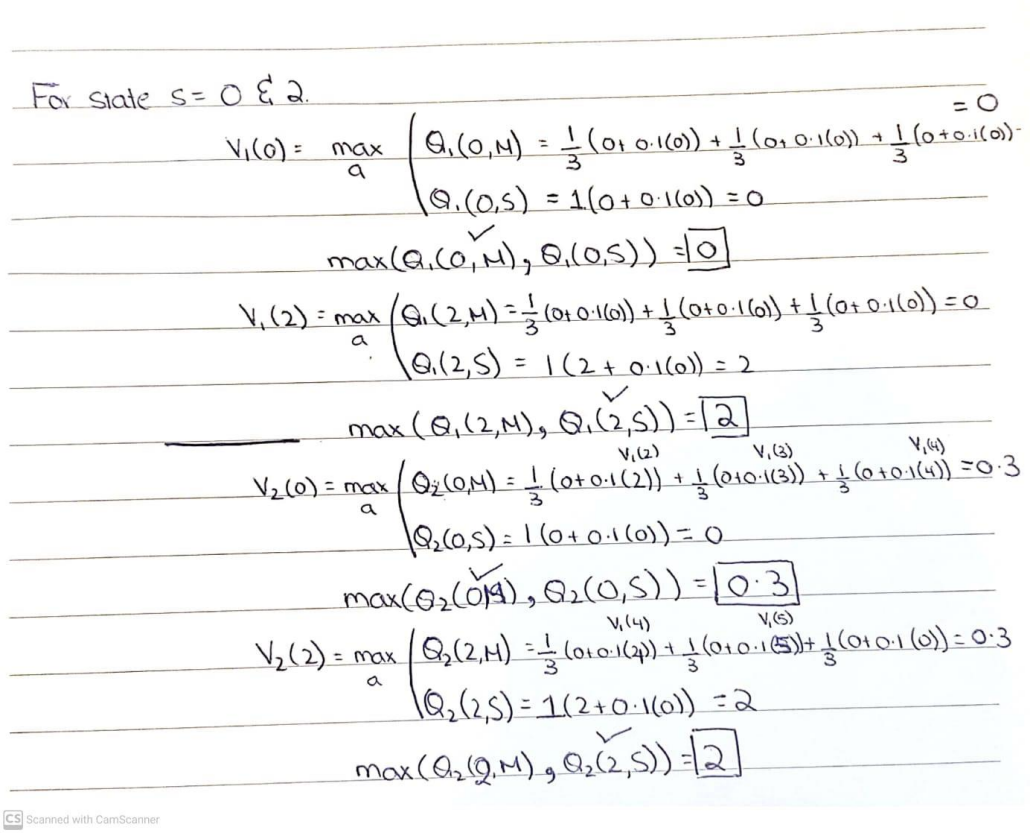
(d) If the value iteration has converged above, write down the optimal policy π*

States	0	2	3	4	5
π^*	Move	Move	Stop	Stop	Stop

(e) How would our results change with γ = 0.1?

Theoretically when we see the discount factor decrease then the model cares more about rewards right now than the future. We say that the model is living in the moment. This has an effect to make our model greedier for short term rewards than long term rewards. In this example the same thing can be observed when we decrease the discounting factor from 1 to 0.1.

For states 3,4 and 5 we can see that these states have an optimal policy of *Stop* which means their value is dependent on the reward which is equal to the state they are in. As the discounting factor simply multiplies with v(Done) = 0, decreasing γ won't make a difference to its results. So, we will look at the calculations for state 0 and 2 below.



From the picture above which shows only 2 iterations of value iteration for states 0 and 2 we can see that with the discounted factor now 0.1, $Q_2(2, M) = 0.3$ whereas when the discounted factor was 1, $Q_2(2, M) = 3$. This shows that the maximum will be $Q_2(2, S)$ which will equal 2 because $Q_X(2, M)$ will always be less than $Q_X(2, S)$ at convergence because it gets multiplied with that 0.1 factor. Thus, S (Stop) will in this be the optimal policy for state 2.

As for state 0, since the Reward for state 0 = 0. Thus $Q_X(0, S)$ will never outperform $Q_X(0, M)$ at convergence since $Q_X(0, M)$ will be greater than 0 as shown by proof in the picture above. Thus, the optimal policy will in this case be M (Move).

Due to these finding we can now tell with 0.1 discount factor the optimal policy for the states will be as follows: {0: Move, 2: Stop, 3: Stop, 4: Stop, 5: Stop}

(f) Perform two iterations of policy iteration for one step of this MDP, starting from the fixed policy below and using (initial) γ=1.0

Assuming **convergence** for Policy Evaluation is when all states in two mini-iterations are same. Both took 3 mini-iterations to converge as shown in the table and calculations below.

Policy Evaluation (Step 1)

Initialize $V_0^{\pi_0} = \vec{0}$

$V_1^{\pi_0}$

M $V_1^{\pi_0}(0) = \frac{1}{3}(0+1(0)) + \frac{1}{3}(0+1(0)) + \frac{1}{3}(0+1(0)) = \boxed{0}$

S $V_1^{\pi_0}(2) = 1(2+1(0)) = \boxed{2}$

M $V_1^{\pi_0}(3) = \frac{1}{3}(0+1(0)) + \frac{2}{3}(0+1(0)) = \boxed{0}$

S $V_1^{\pi_0}(4) = 1(4+1(0)) = \boxed{4}$

M $V_1^{\pi_0}(5) = \frac{1}{3}(0+1(0)) = \boxed{0}$

$V_2^{\pi_0}$

M $V_2^{\pi_0}(0) = \frac{1}{3}(0+1(\overset{V_1^{\pi_0}(2)}{2})) + \frac{1}{3}(0+1(0)) + \frac{1}{3}(0+1(\overset{V_1^{\pi_0}(4)}{4})) = \boxed{2}$

S $V_2^{\pi_0}(2) = 1(2+1(0)) = \boxed{2}$

M $V_2^{\pi_0}(3) = \frac{1}{3}(0+1(0)) + \frac{2}{3}(0+1(0)) = \boxed{0}$

S $V_2^{\pi_0}(4) = 1(4+1(0)) = \boxed{4}$

M $V_2^{\pi_0}(5) = 1(0+1(0)) = \boxed{0}$

$V_3^{\pi_0}$

M $V_3^{\pi_0}(0) = \frac{1}{3}(0+1(\overset{V_2^{\pi_0}(2)}{2})) + \frac{1}{3}(0+1(0)) + \frac{1}{3}(0+1(\overset{V_2^{\pi_0}(4)}{4})) = \boxed{2}$

S $V_3^{\pi_0}(2) = 1(2+1(0)) = \boxed{2}$

M $V_3^{\pi_0}(3) = \frac{1}{3}(0+1(0)) + \frac{2}{3}(0+1(0)) = \boxed{0}$

S $V_3^{\pi_0}(4) = 1(4+1(0)) = \boxed{4}$

M $V_3^{\pi_0}(5) = 1(0+1(0)) = \boxed{0}$

Policy Improvement (Step 1)

$\pi_1(0) = \underset{a}{\operatorname{argmax}} \begin{cases} Q_1(0,M) = \frac{1}{3}(0+1(\overset{V^{\pi_0}(2)}{2})) + \frac{1}{3}(0+1(0)) + \frac{1}{3}(0+1(\overset{V^{\pi_0}(4)}{4})) = 2 \\ Q_1(0,S) = 1(0+1(0)) = 0 \end{cases}$

$\operatorname{argmax}(Q_1(0,M), Q_1(0,S)) = 2 \approx \boxed{M}$

$\pi_1(2) = \underset{a}{\operatorname{argmax}} \begin{cases} Q_1(2,M) = \frac{1}{3}(0+1(\overset{V^{\pi_0}(4)}{4})) + \frac{1}{3}(0+1(\overset{V^{\pi_0}(0)}{0})) + \frac{1}{3}(0+1(0)) = 4/3 \\ Q_1(2,S) = 1(2+1(0)) = 2. \end{cases}$

$\operatorname{argmax}(Q_1(2,M), Q_1(2,S)) = 2 \approx \boxed{S}$

$\pi_1(3) = \underset{a}{\operatorname{argmax}} \begin{cases} Q_1(3,M) = \frac{1}{3}(0+1(0)) + \frac{2}{3}(0+1(\overset{V^{\pi_0}(5)}{0})) = 0 \\ Q_1(3,S) = 1(3+1(0)) = 3 \end{cases}$

$\operatorname{argmax}(Q_1(3,M), Q_1(3,S)) = 3 \approx \boxed{S}$

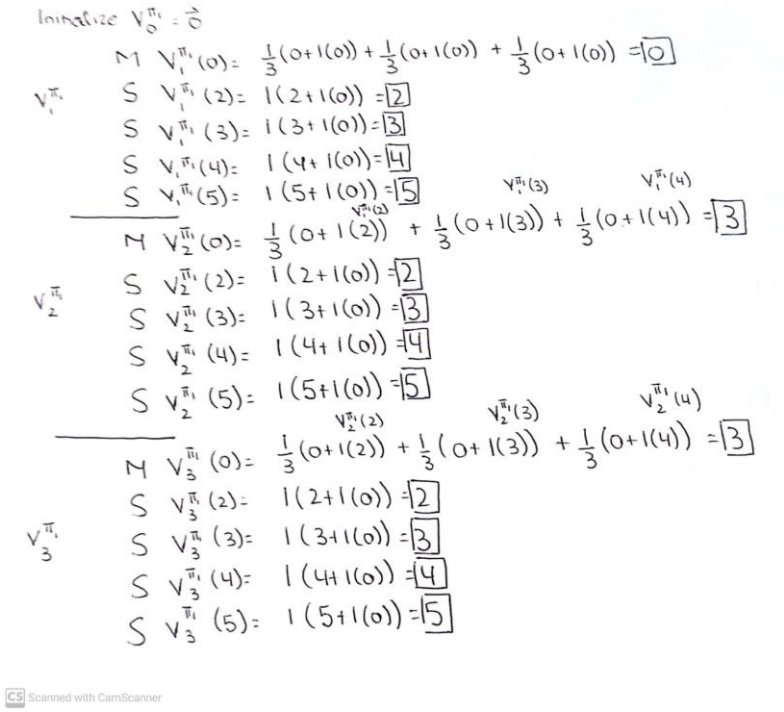
$\pi_1(4) = \underset{a}{\operatorname{argmax}} \begin{cases} Q_1(4,M) = 1(0+1(0)) = 0 \\ Q_1(4,S) = 1(4+1(0)) = 4 \end{cases}$

$\operatorname{argmax}(Q_1(4,M), Q_1(4,S)) = 4 \approx \boxed{S}$

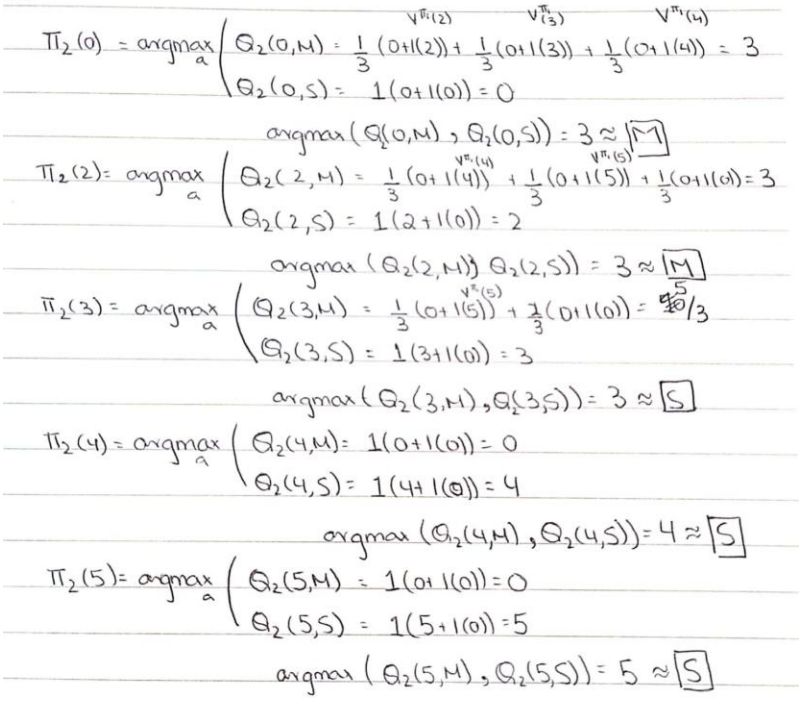
$\pi_1(5) = \underset{a}{\operatorname{argmax}} \begin{cases} Q_1(5,M) = 1(0+1(0)) = 0 \\ Q_1(5,S) = 1(5+1(0)) = 5. \end{cases}$

$\operatorname{argmax}(Q_1(5,M), Q_1(5,S)) = 5 \approx \boxed{S}$

Policy Evaluation (Step 2)



Policy Improvement (Step 2)



States	0	2	3	4	5	
π^0	Move	Stop	Move	Stop	Move	
V^{π_0} 0	0	0	0	0	0	
1	0	2	0	4	0	
2	2	2	0	4	0	] converge
3	2	2	0	4	0	
π^1	Move	Stop	Stop	Stop	Stop	
V^{π_1} 0	0	0	0	0	0	
1	0	2	3	4	5	
2	3	2	3	4	5	] converge
3	3	2	3	4	5	
π^2	Move	Move	Stop	Stop	Stop	

Reinforcement Learning

a) Why is temporal difference learning of q-values (Q-learning) superior to temporal difference learning of values?

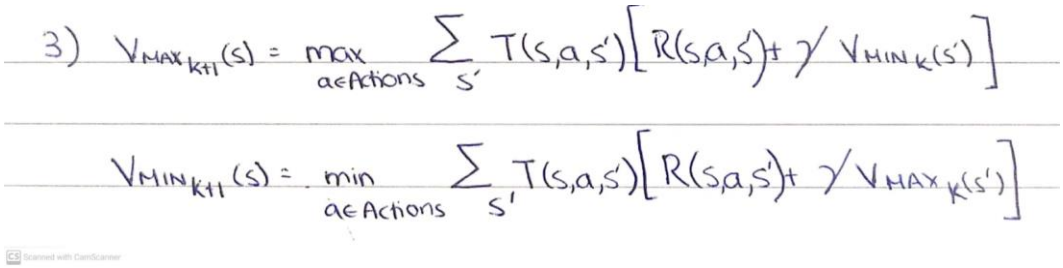
In Q-learning or temporal difference learning of Q-values the policy can simply found by taking the argmax across all actions **a** for **Q(s,a)** where **s** are states which is superior than temporal difference of learning values because to find the policy from learned values you need the Probability of state **s'** given state **s** and action **a** denoted a **P(s'|a,s)** or also known as the Transition model **T(s,a,s').** However, sometimes these transitions aren't available thus making Q-learning more superior as it doesn't have such a limitation.

b) Write a Bellman update for q-value iteration, which is like value iteration except q-values rather than values are learned from previous q-value estimates, using a one-step look ahead (i.e., you should express Qi+1 estimates in terms of Qi estimates).

$$2) \quad Q_{k+1}(s,a) = \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma \max_{a'} Q(s',a') \right]$$

c) In a Markov game, or adversarial MDP, two players, max and min alternate actions in an MDP. Assume the game is zero-sum, so that when a transition (s, a, s´) occurs, max receives R(s, a, s´), while min receives -R(s, a, s´), regardless of who initiated the transition. Assume that both players

have the same set of actions available in any state and that both players use the same discount per time step. Let VMAX(s) and VMIN (s) be the expected future discounted rewards for each player. Write two Bellman equations, expressing each of VMAX and VMIN in terms of adjacent look ahead values of VMAX and / or VMIN.



Max and Min players will both respectively try to maximize or minimize by choosing the actions that works for them. However, additionally since we are assuming it is an adversarial MDP then we know that the discounted utilities will be doing the opposite of what they are trying to do. So, for $VMAX_{K+1}$ we use $VMIN_K$ in our Bellman Equations and vice versa.