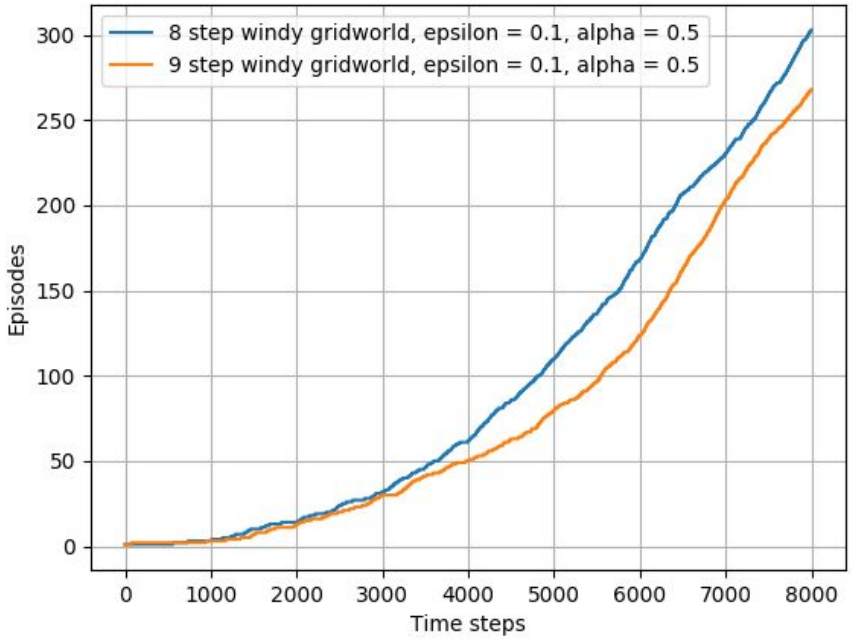
Q1.

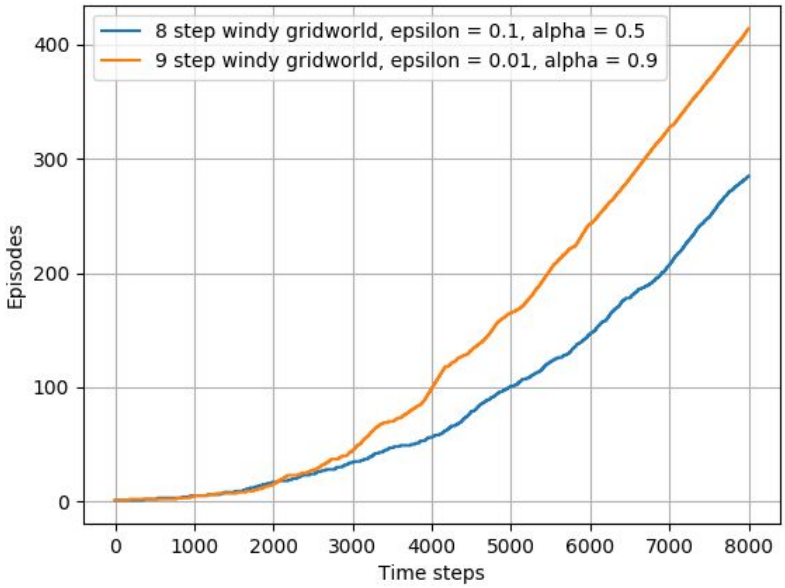
a) We know that regardless of which method we choose, they will converge to the optimal policy. Thus, the final result will be the same. The difference is that the every-visit MC would update the G value of a state after every visit. This would result in many more G values per state. Thus, when the next episode happens, the update of that state would happen a lot more gradually, because it is the average of all values. Thus, the optimal solution wouldn't be found as quickly, and thus first visit MC would converge to the optimal policy much more quickly.

b) This random walk problem can be solved with any of the methods that use markov decision processes to find out the value estimates. Thus, TD(0) and dynamic programming are both different methods that could be used to solve this problem. With dynamic programming you simply just compare each state with the value estimates of the state it could choose, and repeat until the answers converge. With TD(0), we simply just iterate through the task, updating the value estimate. We continue this after a large number of episodes, and the value function should converge. My guess on which algorithm was used to find the true values of ⅙, ⅓, ½, ⅔, ⅚ is dynamic programming, because it can be used very effectively with such a small amount of states, and an easy value estimate.

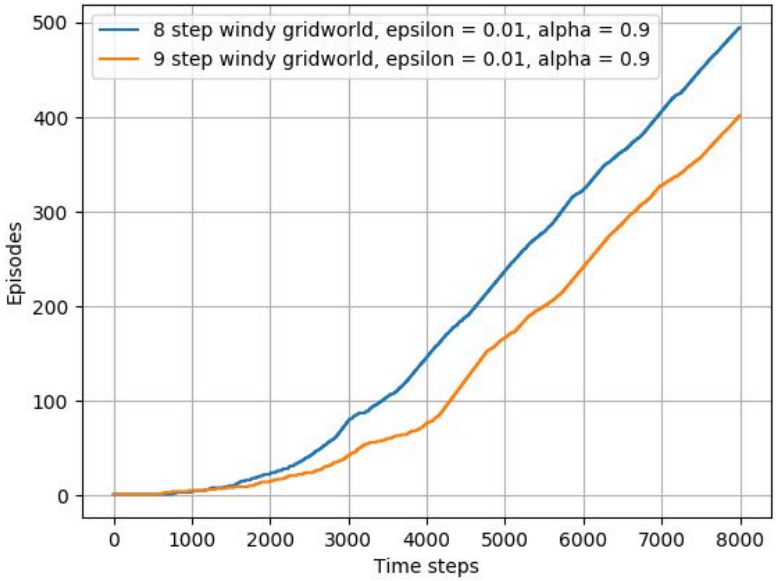
c) When the values of epsilon and alpha are even, the 8-step gridworld outperforms the 9-step gridworld, as seen here:



I was able to make the 9-step gridworld perform much better when I made the epsilon smaller (0.01), and the alpha larger (0.9). This is clear because there are less random steps being made.



When I made the 8 step have the same values, it is the same as seen here:



This is because the best actions to take are the diagonals, and a single downward action. This results in the optimal solution of 7 steps to reach the goal. Thus, having the choice of staying in the same spot doesn't result in a better plot, because it is not part of the optimal solution.

d) Q-learning is off-policy because during the update of a Q(S,A), instead of following the policy and then updating the action value after that, it instead chooses to update the action value based off of the greedy optimal policy. This means that if the policy were to be following an epsilon-greedy approach, the update of a value that takes the epsilon action doesn't depend on the epsilon policy, and instead updates based on the optimal policy. This means that it is a true off-policy method because it doesn't follow the policy given, but rather updates based off of what it believes the optimal policy should be.

Q2.

1) First-visit MC:

Episode 1:

$G_1 = 1 + 0$ therefore $Blist = [1]$

$G_0 = 0 + 1$ therefore $Alist = [1]$

Episode 2:

$G_0 = 2 + 0$ therefore $Alist = [1, 2]$

Episode 3:

$G_2 = 2 + 0$ but state A is found earlier in the episode, so no update.

$G_1 = 0 + 2$ therefore $Blist = [1, 2]$

$G_0 = 0 + 2$ therefore $Alist = [1, 2, 2]$

Episode 4:

$G_3 = 1 + 0$ but state B is found earlier in the episode, so no update

$G_2 = 0 + 1$ but state A is found earlier in the episode, so no update

$G_1 = 0 + 1$ therefore $Blist = [1, 2, 1]$

$G_0 = 0 + 1$ therefore $Alist = [1, 2, 2, 1]$

Episode 5:

$G_1 = 2 + 0$ therefore $Alist = [1, 2, 2, 1, 2]$

$G_0 = 0 + 2$ therefore $Blist = [1, 2, 1, 2]$

Episode 6:

$G_2 = 1 + 0$ but state B is found earlier in the episode, so no update

$G_1 = 0 + 1$ therefore $Alist = [1, 2, 2, 1, 2, 1]$

$G_0 = 0 + 1$ therefore $Blist = [1, 2, 1, 2, 1]$

Episode 7:

$G_1 = 2 + 0$ therefore $Alist = [1, 2, 2, 1, 2, 1, 2]$

$G_0 = 0 + 2$ therefore $Blist = [1, 2, 1, 2, 1, 2]$

Episode 8:

$G_1 = 1 + 0$ therefore $Blist = [1, 2, 1, 2, 1, 2, 1]$

$G_0 = 0 + 1$ therefore $Alist = [1, 2, 2, 1, 2, 1, 2, 1]$

Episode 9:

$G_0 = 1 + 0$ therefore $Blist = [1, 2, 1, 2, 1, 2, 1, 1]$

Episode 10:

$G_0 = 1 + 0$ therefore $Blist = [1, 2, 1, 2, 1, 2, 1, 1, 1]$

Episode 11:

$G_0 = 1 + 0$ therefore $Blist = [1, 2, 1, 2, 1, 2, 1, 1, 1, 1]$

Episode 12:

$G_3 = 2 + 0$ but state A is found earlier in the episode, so no update

$G_2 = 0 + 2$ but state B is found earlier in the episode, so no update

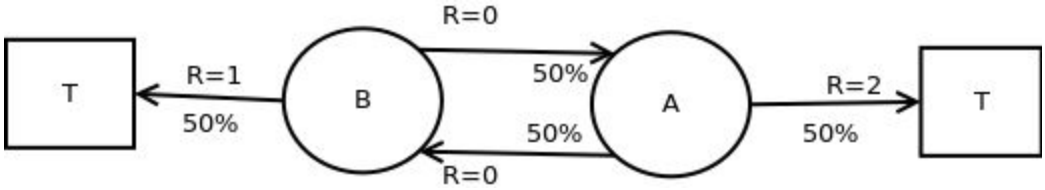
$G_1 = 0 + 2$ therefore $Alist = [1, 2, 2, 1, 2, 1, 2, 1, 2]$

$G_0 = 0 + 2$ therefore $Blist = [1, 2, 1, 2, 1, 2, 1, 1, 1, 1, 2]$

Thus, since V(S)= Avg(returns)

$V(A) = Avg(Alist) = 1.55556$ and $V(B) = Avg(Blist) = 1.363636$

2)



This is the state-transition diagram created. This was created because every time there was a shift from A to B, or B to A, the reward was 0. Thus, the transition between B and A equals a reward of 0. Every time A terminated, the reward was 2, thus I predicted the reward for A to T is 2. Every time B terminated, the reward was 1. Thus, the predicted reward from B to T is 1.

The total number of rewards between all states is roughly equal. That is, there are 6 counted rewards for state A->B, 6 counted rewards for state B->A, 5 counted rewards for A->T and 7 for B->T. Seeing that they are all so close, it is safe to assume that the probabilities of each action is roughly equal, i.e. 50%. Thus, under these probabilities, the markov decision reward model would be:

$V(A) = 0.5 * (2 + 0) + 0.5 * (0 + V(B)) \Rightarrow V(A) = 1 + 0.5V(B)$

$V(B) = 0.5 * (1 + 0) + 0.5 * (0 + V(A)) \Rightarrow V(B) = 0.5 + 0.5(1 + 0.5(V(B))$

$V(B) = 1 + 0.25V(B) = \frac{1}{1-0.25} = 1.3333$

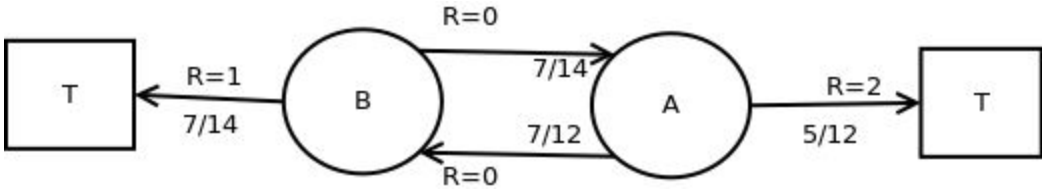
$V(A) = 1 + 0.5(1.3333) = 1.66667$

And thus, this is my maximum-likelihood of model, given these episodes.

3) Seeing as how this is a batch TD method, we can see that unlike the first-visit MC, we will be considering every return state. Thus, we must make a list of all the returns of each state:

$listA = [0, 2, 0, 2, 0, 0, 2, 0, 2, 0, 0, 2]$ and $listB = [1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0]$

We can see that A returned 2 5 times, and B returned 1 7 times. A and B both returned 0 7 times. This means B had 14 actions and A had 12 actions. With this in mind, we can see that the batch TD would setup the markov model to be setup like this:



With this setup, we can see that

$V(B) = \frac{7}{14} * (1 + 0) + \frac{7}{14}(0 + V(A)) \Rightarrow V(B) = \frac{1}{2} + \frac{1}{2}V(A)$

$V(A) = \frac{5}{12} * (2 + 0) + \frac{7}{12}(0 + V(B)) \Rightarrow V(A) = \frac{10}{12} + \frac{7}{12}(\frac{1}{2} + \frac{1}{2}V(A))$

$\Rightarrow V(A) = \frac{10}{12} + \frac{7}{24} + \frac{7}{24}V(A) \Rightarrow \frac{27}{24} + \frac{7}{24}V(A) \Rightarrow V(A) = \frac{\frac{27}{24}}{1-\frac{7}{24}} = 1.588235294$

$V(B) = \frac{1}{2} + \frac{1}{2}V(A) \Rightarrow V(B) = \frac{1}{2} + \frac{1}{2}(1.588235294) = 1.294117647$