Report for Assignment 1 - Artificial Intelligence (CS - F407) by Shaunak Sunil Damle (2021A7PS2607G)

Important Note : I have reused my driver code and have not written a new function for every new graph, so please see the functions and use them accordingly to get the required outputs.

Question 1

1) For **E-Greedy** Approach, the Graph for % of times the optimal reward is selected and the average reward is as follows.

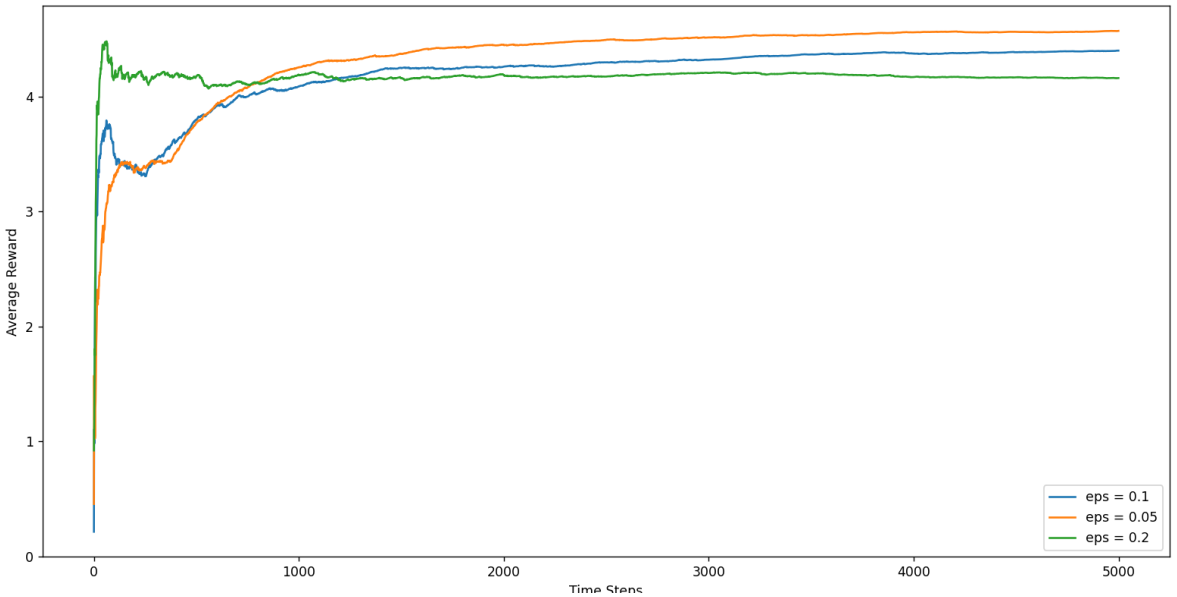


Figure 1

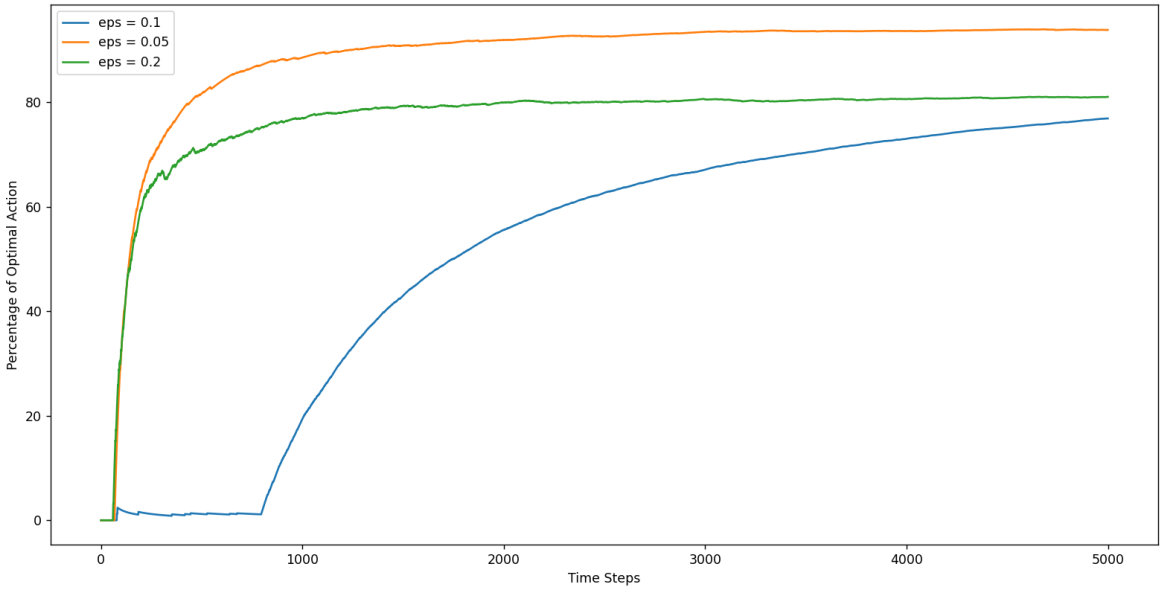


Figure 2

-> Here, we can see in figure 1 that as we reduce value of epsilon, the Average reward increases slowly as compared to a higher epsilon value, but it will get us closer to the best reward for a particular action, but that comes at the price of more number of runs required for the experiment.
-> In figure 2, the initial flat line indicates the selection of an unoptimal action for a large number of timesteps, but eventually the optimal actions were chosen by the algorithm, hence the percentage of optimal action selection increases eventually.
-> Similar to figure 1, a lower epsilon value chooses the best action eventually, but here it chooses the optimal action faster as compared to the other epsilon values, this is randomised so we cannot comment on when which particular epsilon will have better exploratory actions.

2) For **Optimistic Initial Values Approach**, the Graph for % of times the optimal reward is selected and the average reward is as follows.

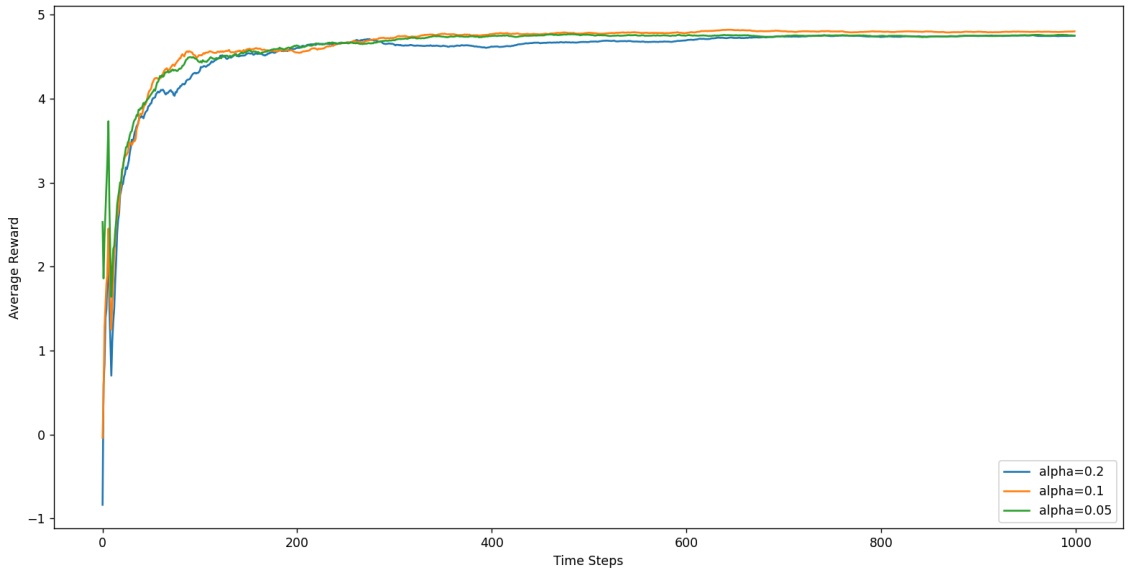


Figure 1

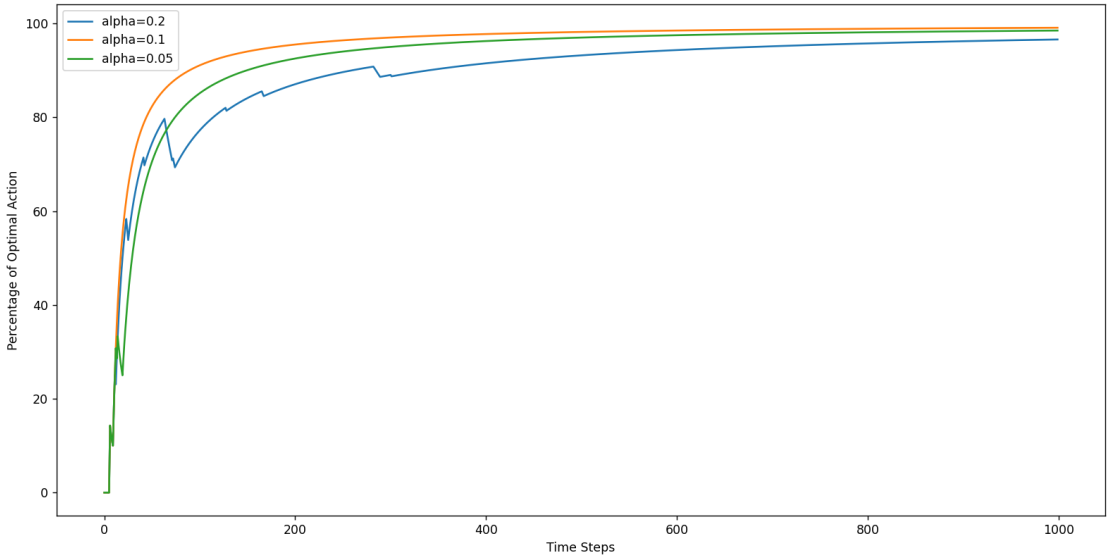


Figure 2

->Here similar to E-greedy, the alpha functions similar to epsilon with values coming closer to alpha with decreasing values of alpha.

3) For Upper-Confidence Bound Approach, the Graph for % of times the optimal reward is selected and the average reward is as follows.

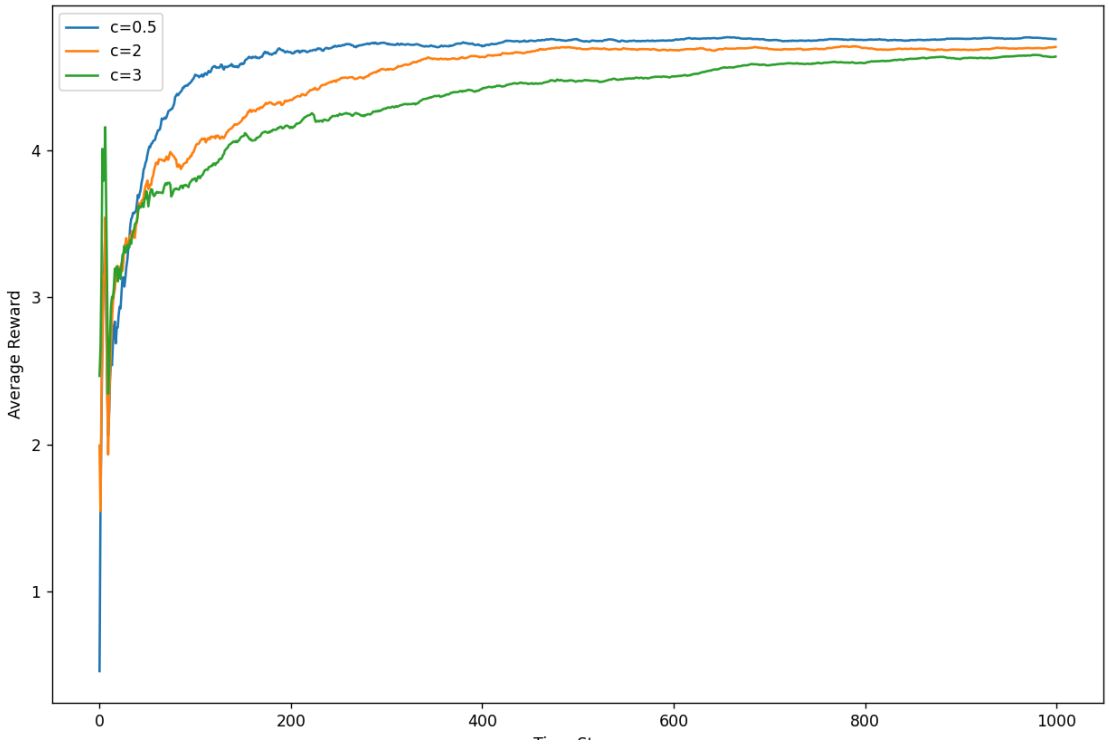


Figure 1

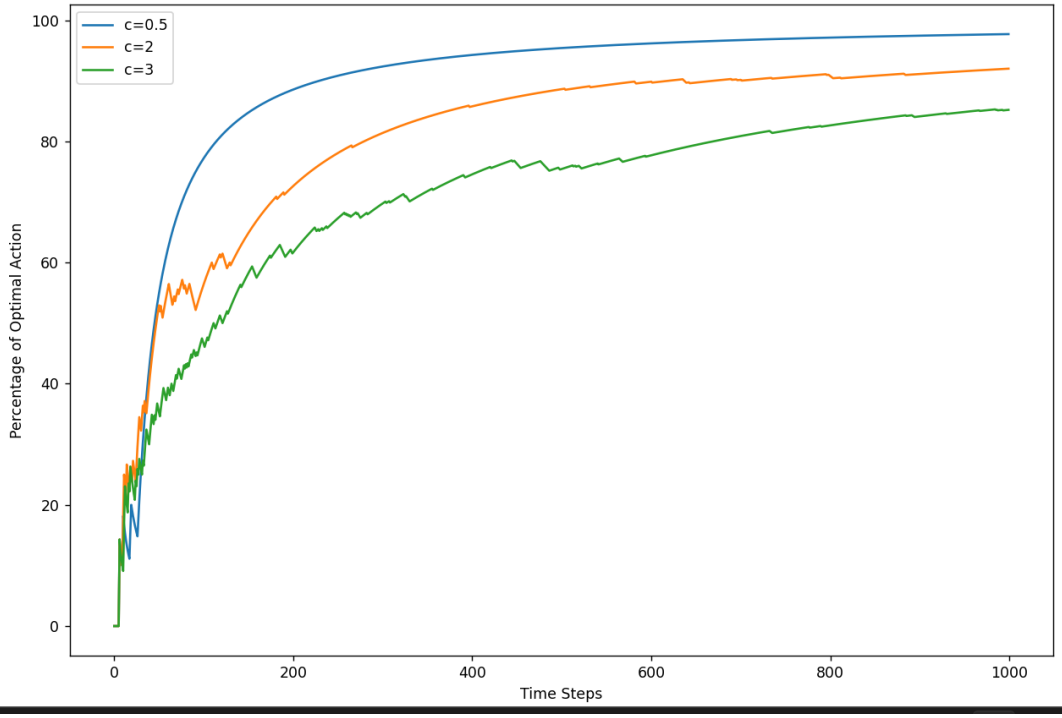


Figure 2

-> Here we can see as the value of c reduces, we get closer to the true optimal action, but we have considered only 1000 time steps here, we can clearly see that for

c=2 and c=3, the functions are having a steeper slope hence they are still growing whereas for c=0.5 we have reached a plateau.

-> Higher the value of c, more exploratory our algorithm is, hence as the number of time steps increases, our algorithm comes much closer to the true optimal action value as compared to a smaller c .

Comparison of all methods :

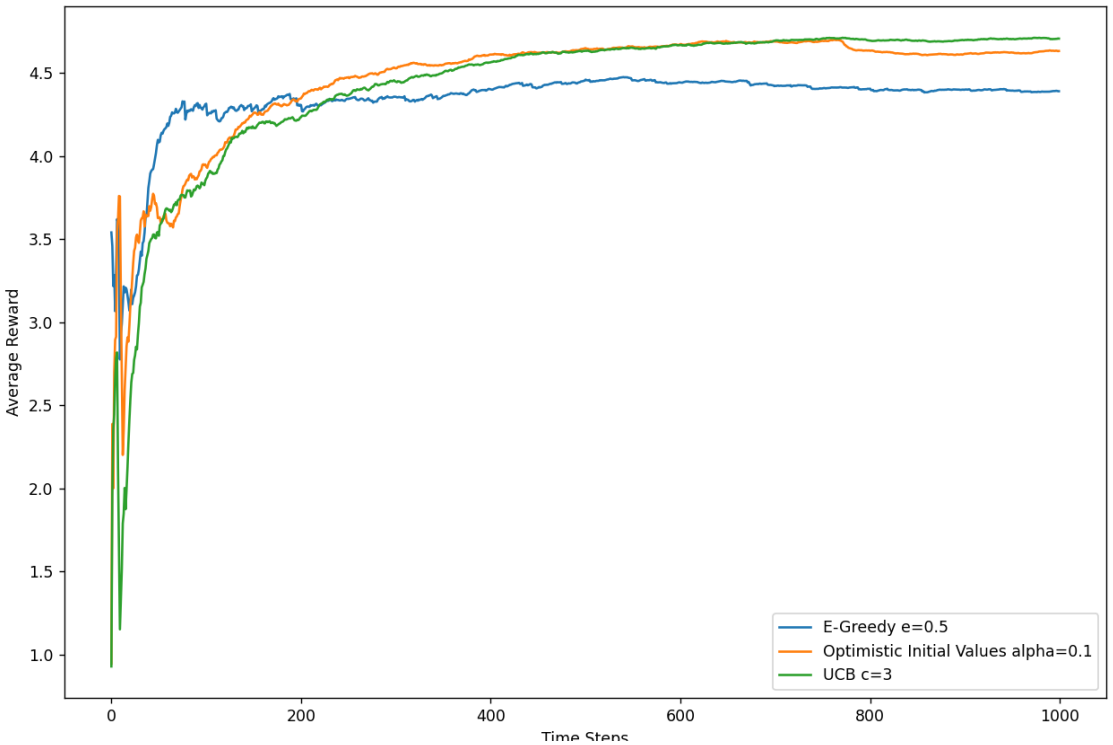


Figure 1

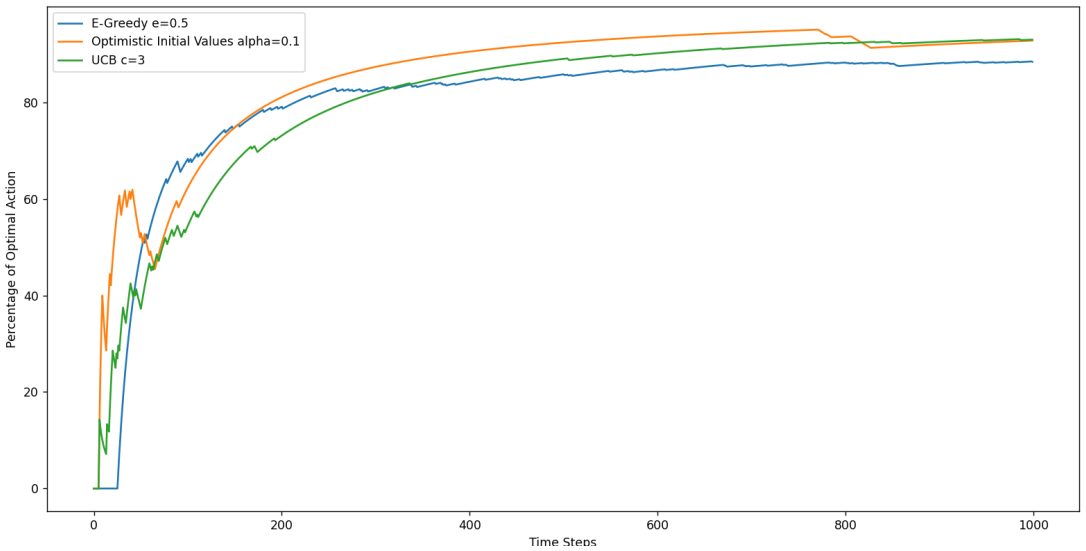


Figure 2

-> For Optimistic Initial Values, we have taken Q value as 5.

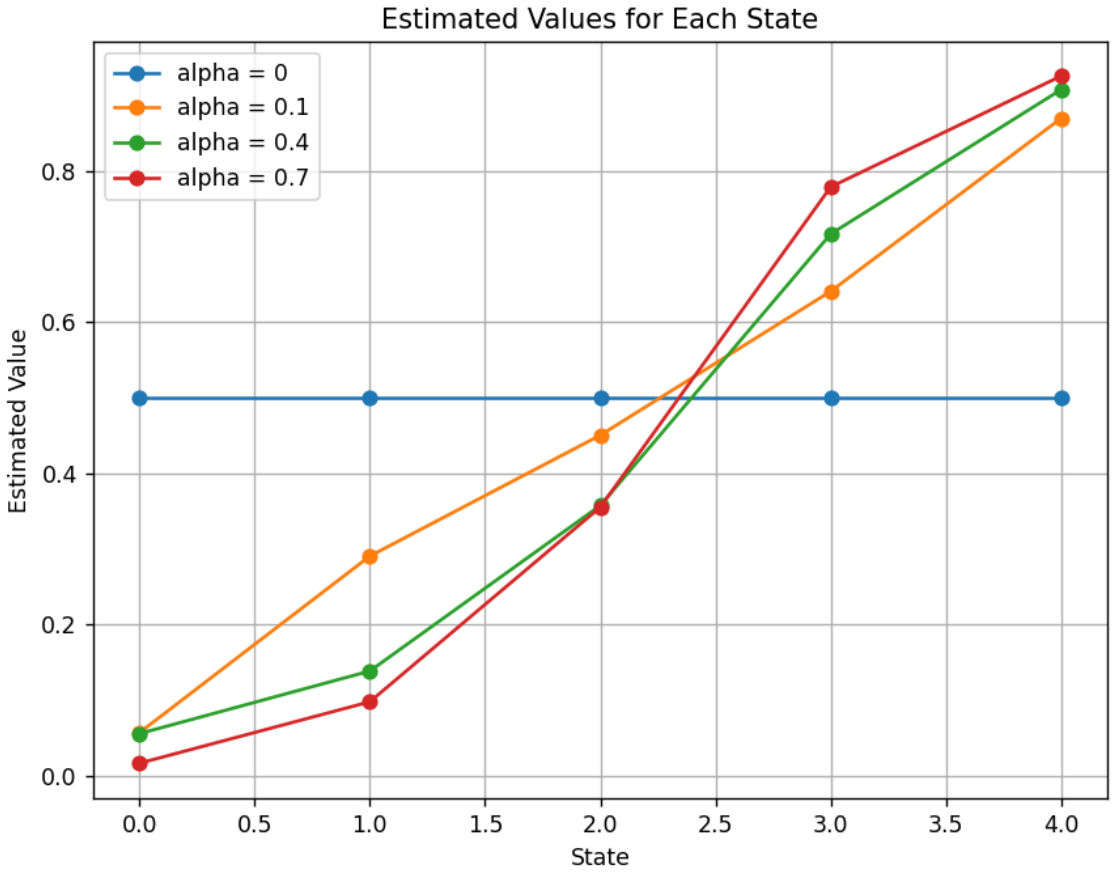
->Here we can clearly see from both figure 1 and 2 E-greedy starts resting at a value which is lesser than the Optimal True action value. As we start from a large initial

value greater than the optimal value for Optimal Initial values approach, we start converging towards a value much closer to the true action value for that function.

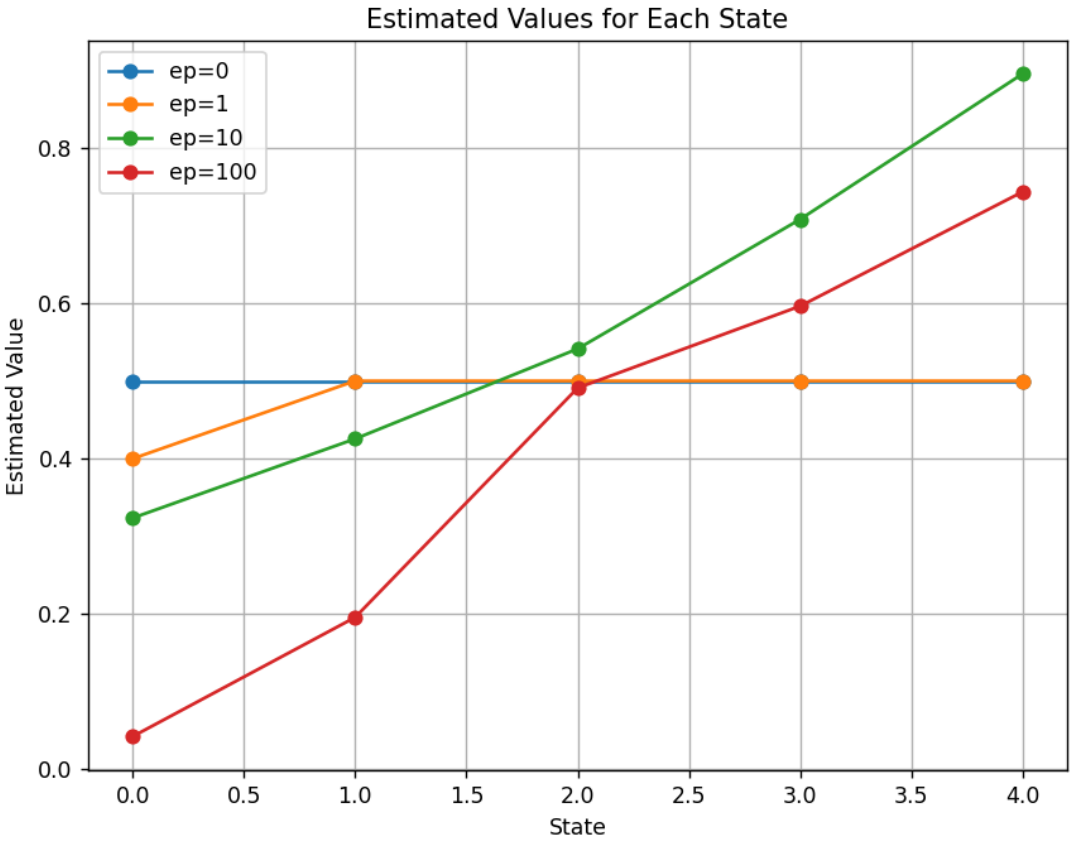
-> The UCB approach gives us the closest approximation to our true action value hence proving to be the best approach out of the three shown above.

Question 2

The graphs below show the difference in results obtained based on the different values of alpha used in the algorithm.



The above graph depicts the different estimated values for various alpha values.



This graph denotes the variation of estimated values with difference in episodes.

Does the root-mean-squared errors converge to zero? Why?
- > No, the RMS errors does not converge to 0 but it converges to a particular value. This is because after a certain point, we will start overshooting the global minima value which will cause us to keep hopping around that particular value giving us a fixed minimum rms value. If we further reduce our alpha values, we will require more episodes but it will give us better estimates reducing our rms error more.

What will happen to the root-mean-squared errors when α = 1/n
- > The root mean squared errors are reduced as using a dynamic alpha that adapts to the number of visits or occurrences of a particular action can improve the estimated values faster than that of a fixed alpha