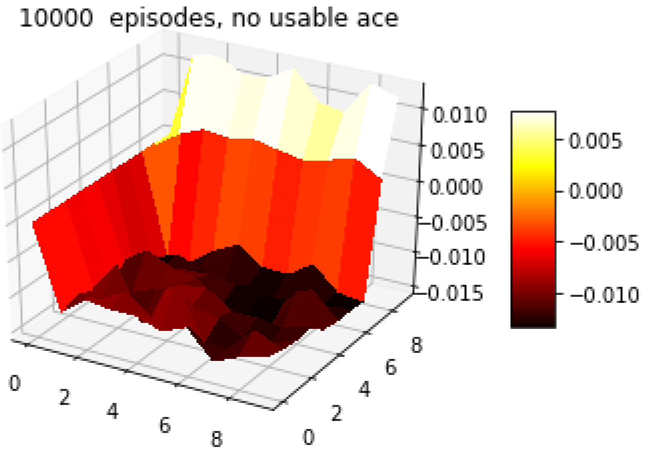
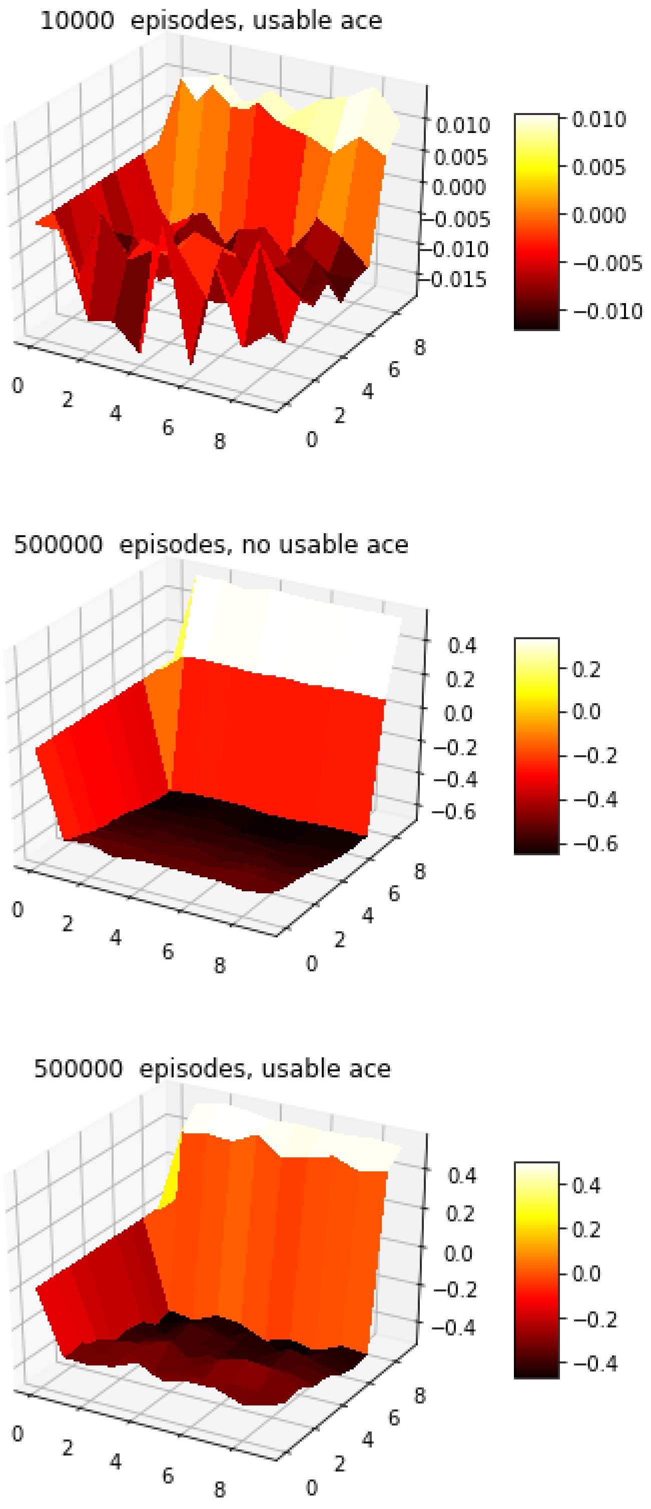
RL HW 3
Figures and Explanations

All theory answers are in the hw3_theory.pdf file.
All codes are in the github repo.

4) Fig 5.1

This code is for MC prediction. We use every visit MC prediction for evaluating the policy provided to us. The policy is that the player sticks at 20 and 21 and hits at every other number. The code first simulates 10000 episodes of the given policy and then continues the simulation for 500000 episodes. The generated value function plots are shown below:





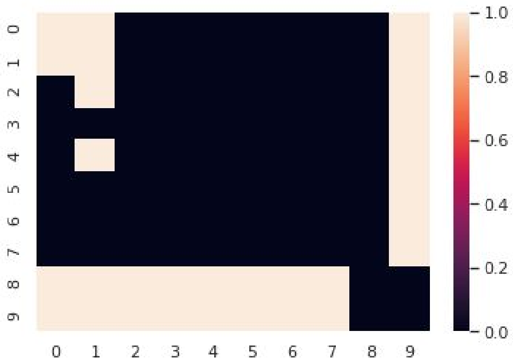
10000 episodes, usable ace

500000 episodes, no usable ace

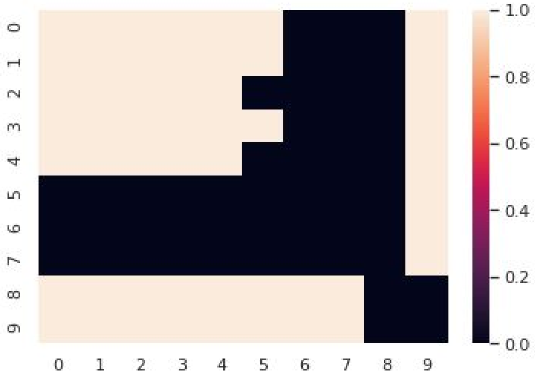
500000 episodes, usable ace

Fig 5.2

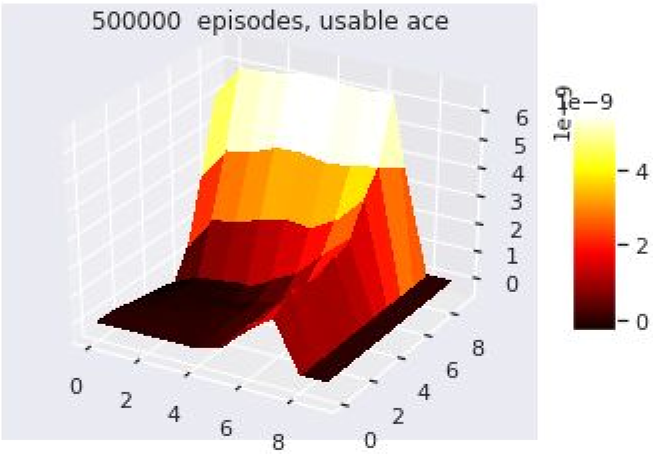
For this part, we had to implement on-policy MC control with exploring starts. In each episode, we first select a random state-action pair with all state-action pairs selected with non zero equal probabilities. Then, we do the action value updation and the optimal policy improvement by selecting greedy actions. The optimal policy that is obtained along with the optimal value function obtained is:



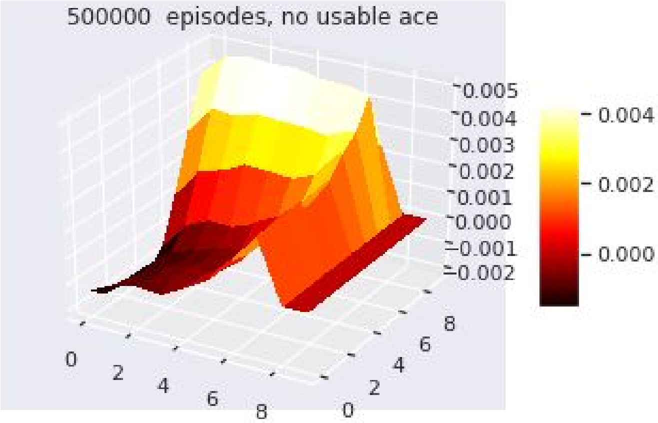
Optimal policy for non usable ace



Optimal policy for usable ace



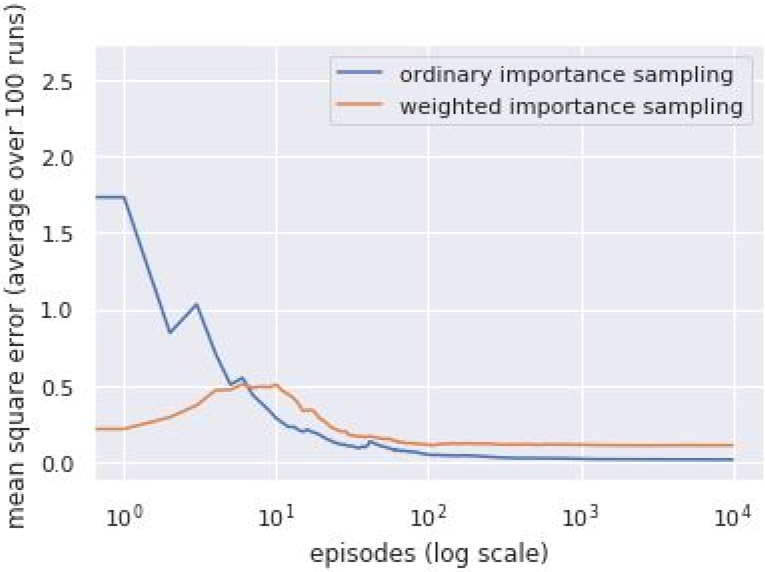
Optimal value function for usable ace



Optimal value function for non usable ace

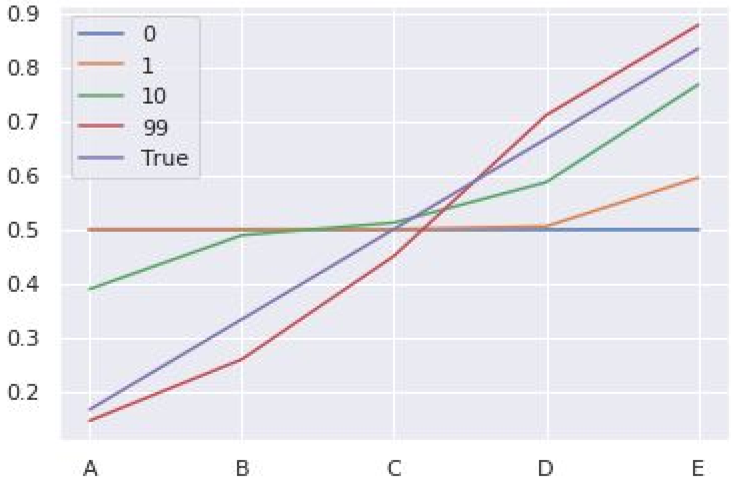
Fig 5.3

Here, we had to implement off-policy MC control with a given stochastic behaviour policy and a given deterministic target policy. Then, we compare the rewards obtained between the ordinary importance sampling and weighted importance sampling. We can clearly see from the plot that weighted importance sampling produces better estimates i.e. lower error estimates of the value function for the off-policy behaviour.

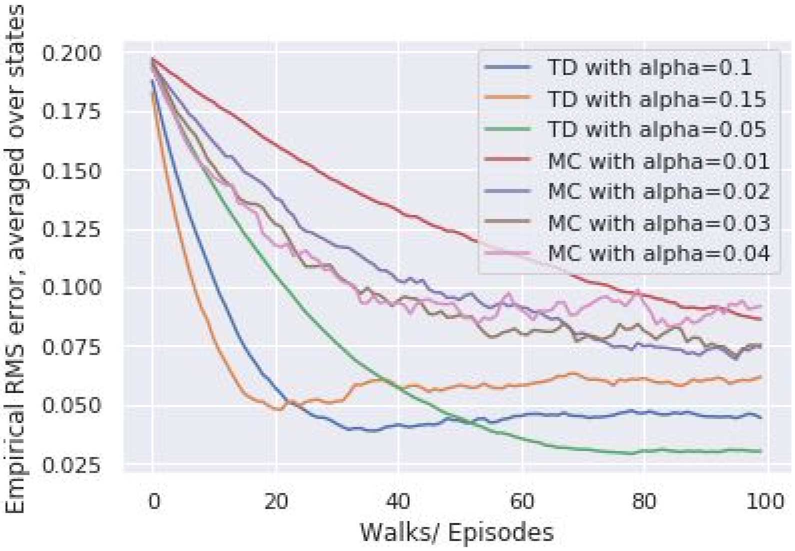


6) Eg. 6.2

In this question, we had to compare the relative root mean square error values obtained for TD(0) and MC-alpha over different alpha values. The plots are:



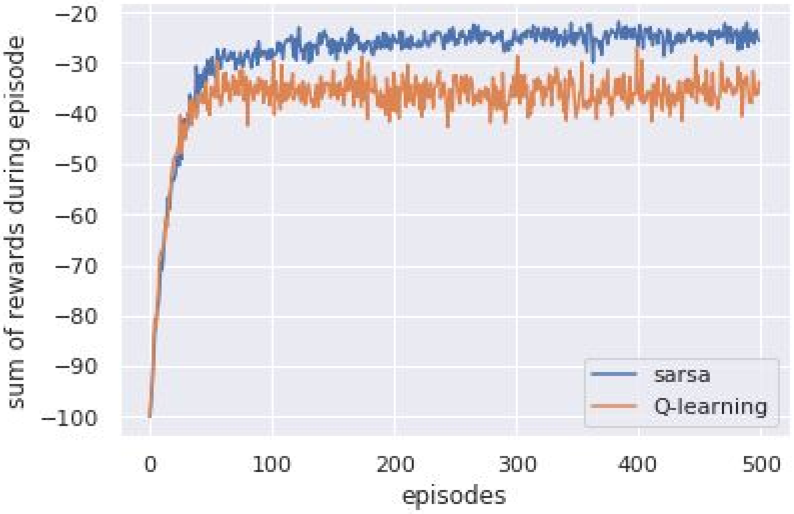
This graph shows the values learned after different numbers of episodes for one run of TD(0). The estimates after 100 episodes are very close to the true values.



This plot shows learning curves for the TD(0) and MC-alpha over different alpha values. We compare RMSE values (computed between the value function learned and true value function, averaged over the five states). We initialise the value function estimates to 0.5 for all states. We clearly see that TD is consistently better than the MC for this particular task

7) Eg. 6.6

In this question we compare the performance of both SARSA and Q-learning on the cliff walking problem. We see that the Q-learning algo learns the risky but efficient optimal path to reach goal state but the SARSA algo learns the safer but longer path. Since, Q-learning is more prone to falling off the cliff due to risky behaviour, its sum of rewards tends to be lower. However, due to the safe path learnt by SARSA, the sum of rewards is higher on average. The plot is:


sum of rewards during episode
episodes
sarsa
Q-learning