Part 1.1

Learning rate : α(N)=C/(C+N(s,a))

C = 10

We set C = 10 so the learning rate will be always between 0 and 1, and for unexplored state, the learning rate will be equal to 1 and as the number of exploration of state increases, the learning rate will decrease. So for unexplored states or states that are not explored frequently, the learning rate is high so the learning process will be fast, and for states that have been frequently visited, the learning process will be slow.

discount factor: 0.9

The discount factor has to be between 0 and 1. Set the discount factor to 0.9 since this will consider the long term reward.

Ne: set the Ne to 5

To learn an optimal policy for Q learning, we need the exploration function so that the agent can further explore on top of the existing explored states. Set the Ne of exploration function to 5, which is considered frequent state in the training process, so we can explore new state after we have seen the explored states over 5 times.

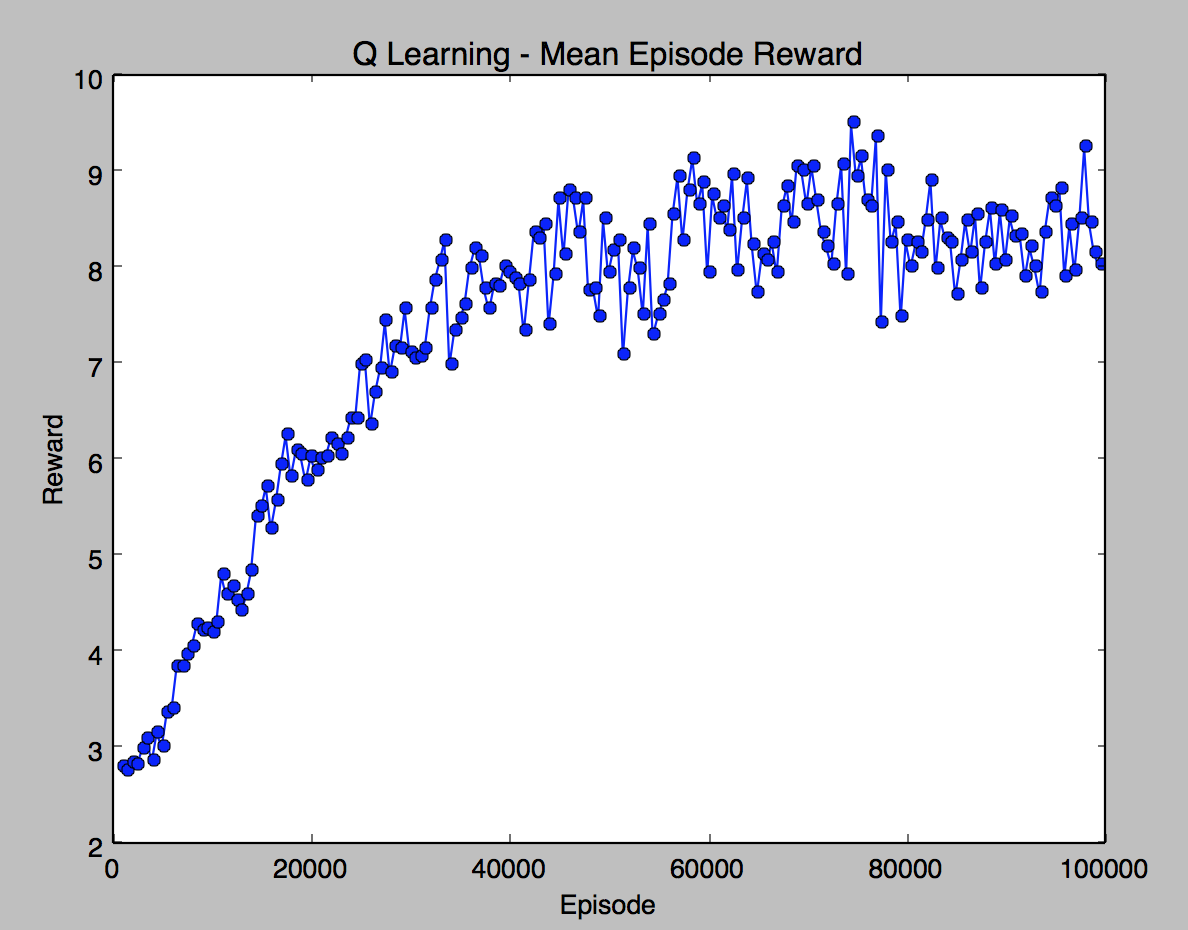
Games needed: about 100000 game is needed to learn an optimal policy

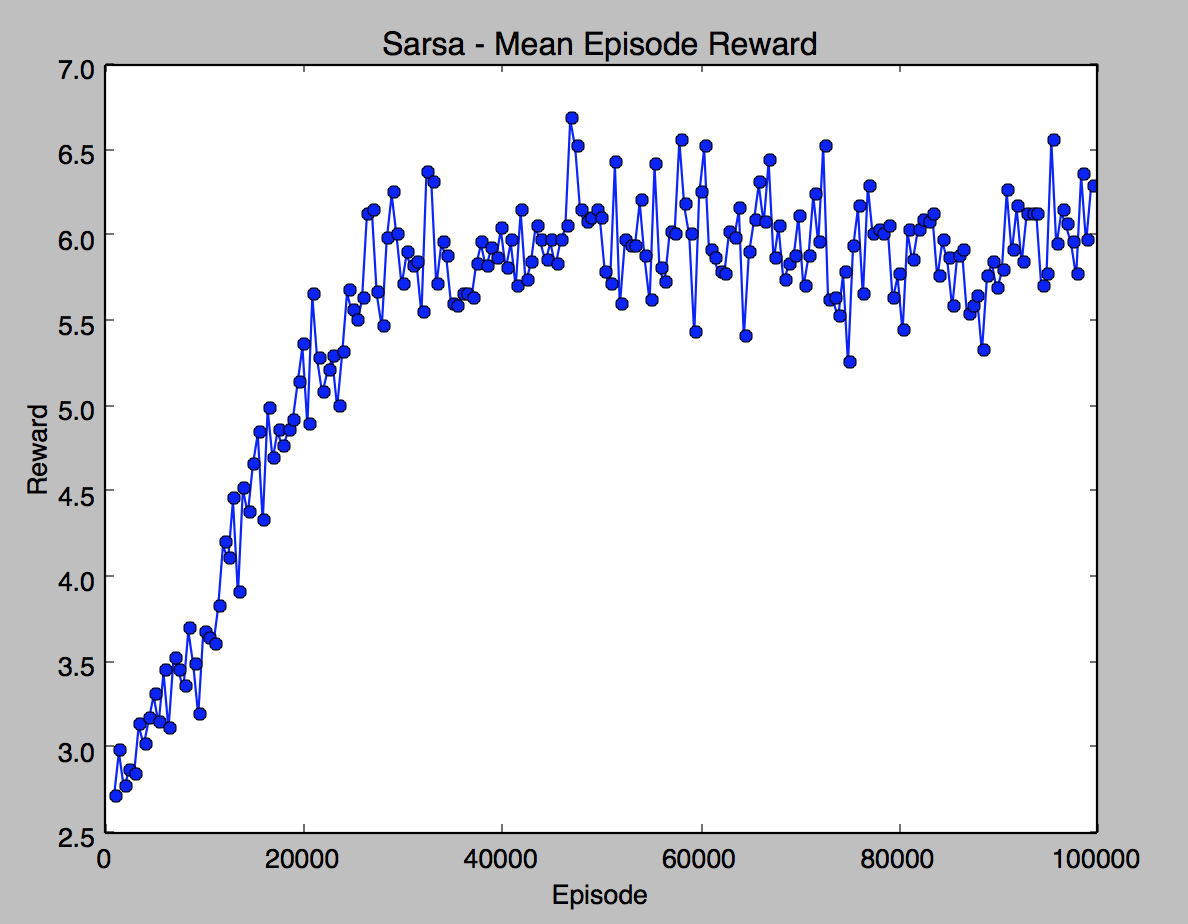
TD Average number of bounces: about 9 – 10

Sarsa Average number of bounces: about 6 – 7

Plot:

TD:



SARSA: 

File: final\_pong.py sarsa.py