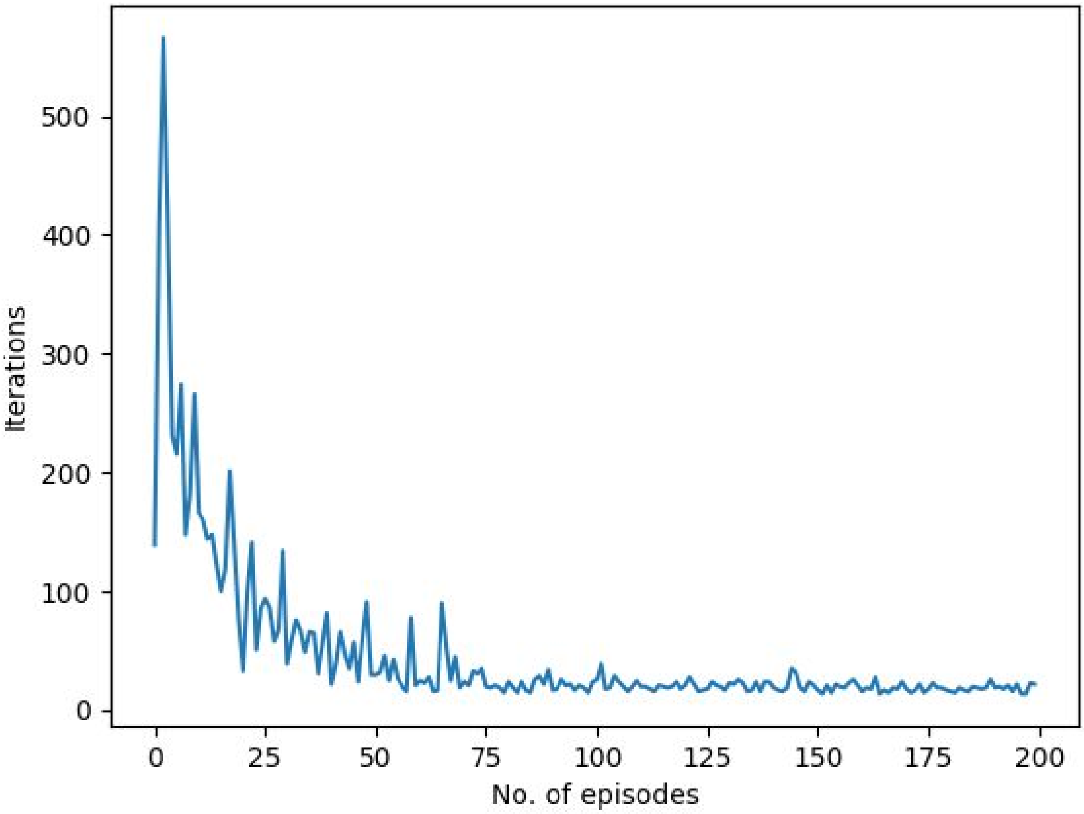
Assignment 4
Report
Lakshya Bansal(2016240)

Q3 a) For this part, a random exploration factor *epsilon = 0.3* , learning rate *alpha = 0.1*, and discount factor *gamma = 0.4* is taken. Epsilon is taken to incorporate a random exploration factor to avoid the looping of the agent in a local area, i.e, the agent might get stuck in an area since the Q values in for certain actions in a state might force the agent to take an action which is not optimal. The learning curve obtained is shown :



With the passage of episodes, the agent slowly learns the optimal actions at every state, and hence, the number of iterations to reach the goal state decrease.

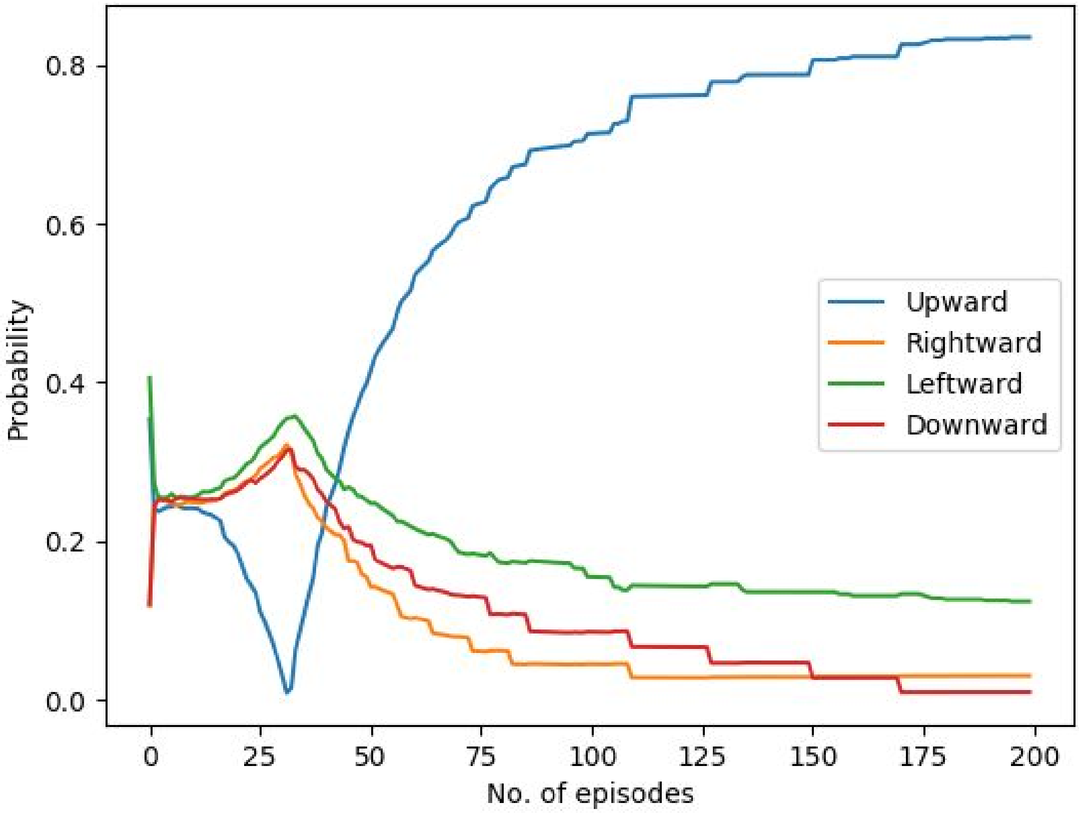
b) For this part, the code was modified to print the number of consecutive times the agent reached the goal state within 50 steps. These evaluations are done on a fixed environment.

Alpha	Gamma	Number of episodes
0.1	0.4	119
0.4	0.4	176
0.8	0.4	184
1	0.4	187
0.1	0.1	20
0.4	0.1	140
0.8	0.1	180
1	0.1	182
0.1	0.8	150
0.4	0.8	185
0.8	0.8	188
1	0.8	191

Alpha determines to how much extent the newly learnt values override the previous values in the Q table. A low alpha makes the agent learn less from the actions it takes, while a high alpha makes the agent learn fast. In the above table, for every different value of gamma, high alpha gives better results. It is expected since, being a deterministic environment, *alpha = 1* should give better results.

Gamma determines how much weightage the agent gives to previous rewards(previous Q table values). A low gamma means that the agent considers only the current reward and ignores all previous rewards. A high gamma means that the agent looks for an overall high reward rather than giving priority to high rewards. A *gamma = 1* means that the agent is never satisfied and the algorithm will never converge. In the above table, the gamma = 0.8 gives better results and this is also expected since the agent appropriately considers both the current and previous rewards.

c)



In this part, at every step, the action is chosen probabilistically. Initially, the Q table is initialised with all zeros, hence each action has equal probability of 0.25. At every step, an action receives a reward which is updated using the bellman equation of temporal difference Q Learning. After certain number of episodes, the Q value of an action is higher than the other actions, and hence the probability that the action is chosen is high, which is also verified by the graph above. The state is observation is *(3,2).*
In the state *(3,2)* , since the action 'up' is the optimal action to reach the goal state(observed by human eye), it is expected that the upward action should have the highest probability which is also visible in the graph.

d) The best alpha and beta as observed in 'a)' are :

Alpha = 1

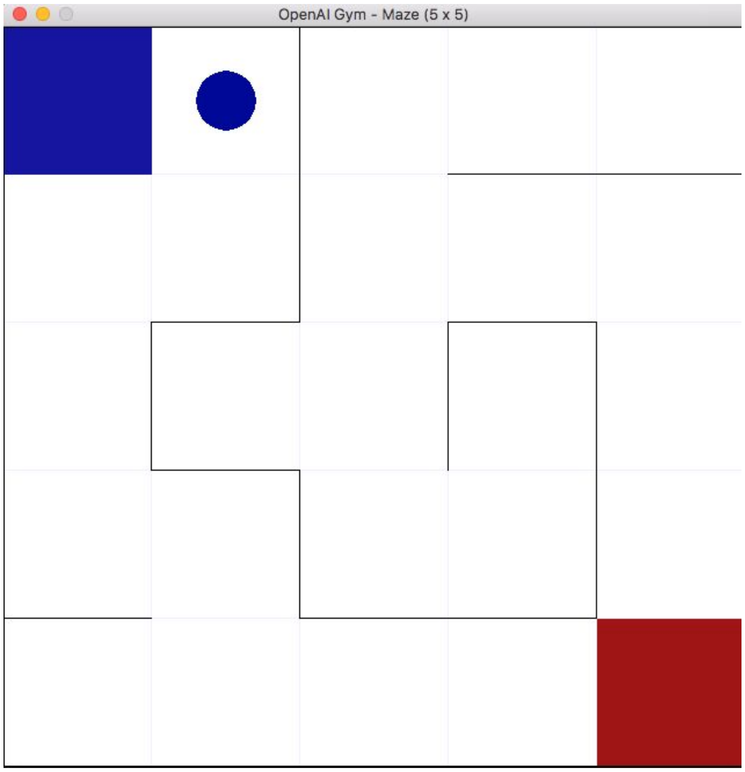
Gamma = 0.8

We will vary the epsilon(random exploration parameter) and analyse the result.

Alpha	Gamma	Epsilon	No. of episodes
1	0.8	0.3	186
1	0.8	0.5	50
1	0.8	0.1	192
1	0.8	0.7	4

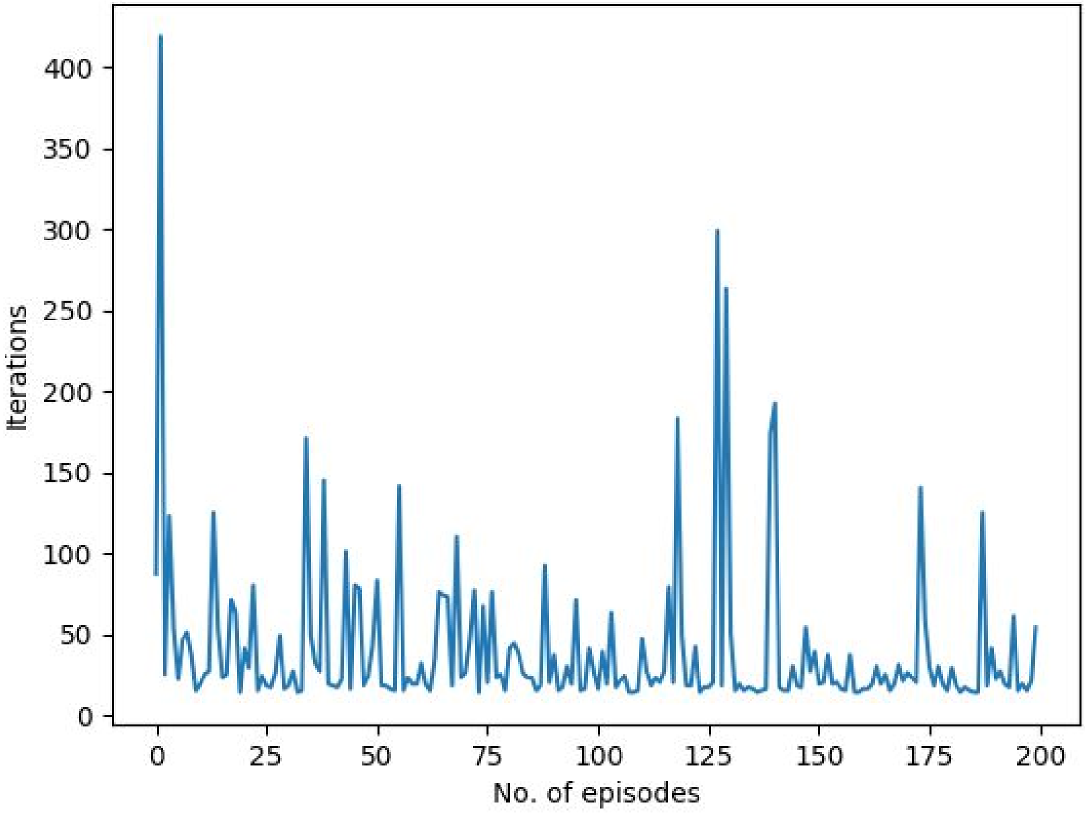
e) The exploration factor gamma determines how frequently a random action has to be taken. A high gamma means that the agent will take a random action more frequently, and hence is expected to reach the goal state after more number of iterations. A low gamma means that the agent follows the Q table more and explores the environment less frequently, hence it is expected that the agent learns and reaches the goal states quickly, as can be seen from the table above.

Q4. When the trained agent (Q table of the previous maze) was implemented for a new maze, the agent got stuck at a particular state. This was expected since optimal actions for a new maze would be different.



A way to solve this is that random exploration can be incorporated to the Q table and the Q table can be updated with the Bellman equation. Doing this it was noted that the agent converged for the Q table quickly and was able to reach the goal state.

BONUS :



As can be seen from the graph above, the algorithm starts finding an optimal path quicker, i.e., within 5 steps. The various perturbations in the graph is due to the randomness involved in the algorithm due to the epsilon factor.