COL864
Assignment2

Akarsh Sharma
2018EE10435
Aniket Shetty
2018EE10443

May 5, 2021

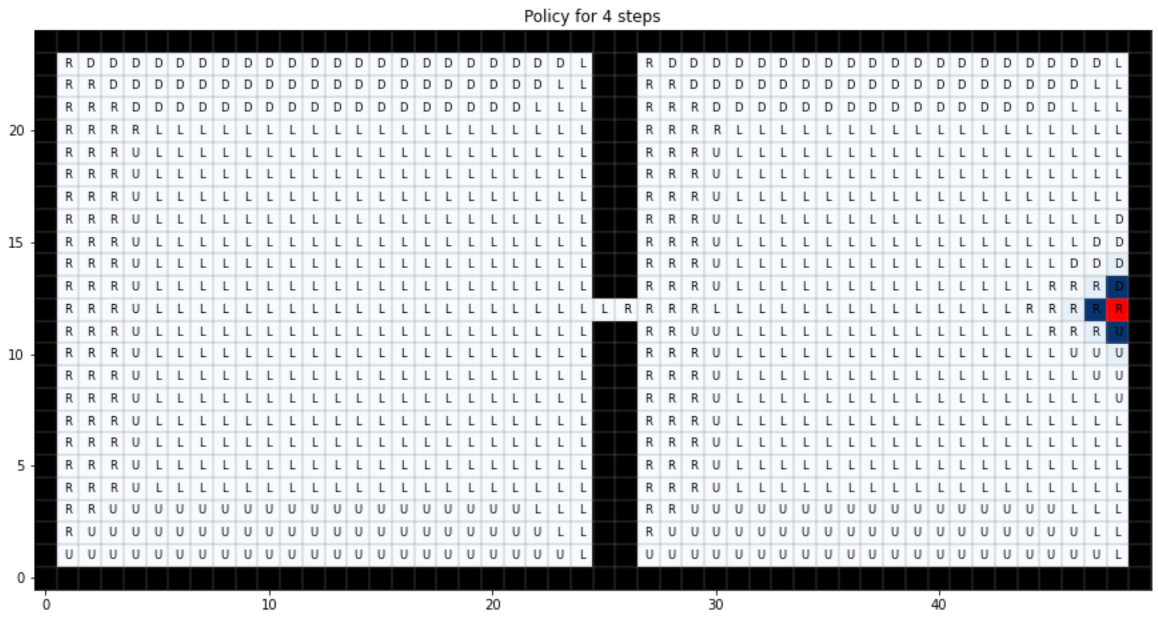
1 Question 1 - Solving an MDP

1.1 Overview

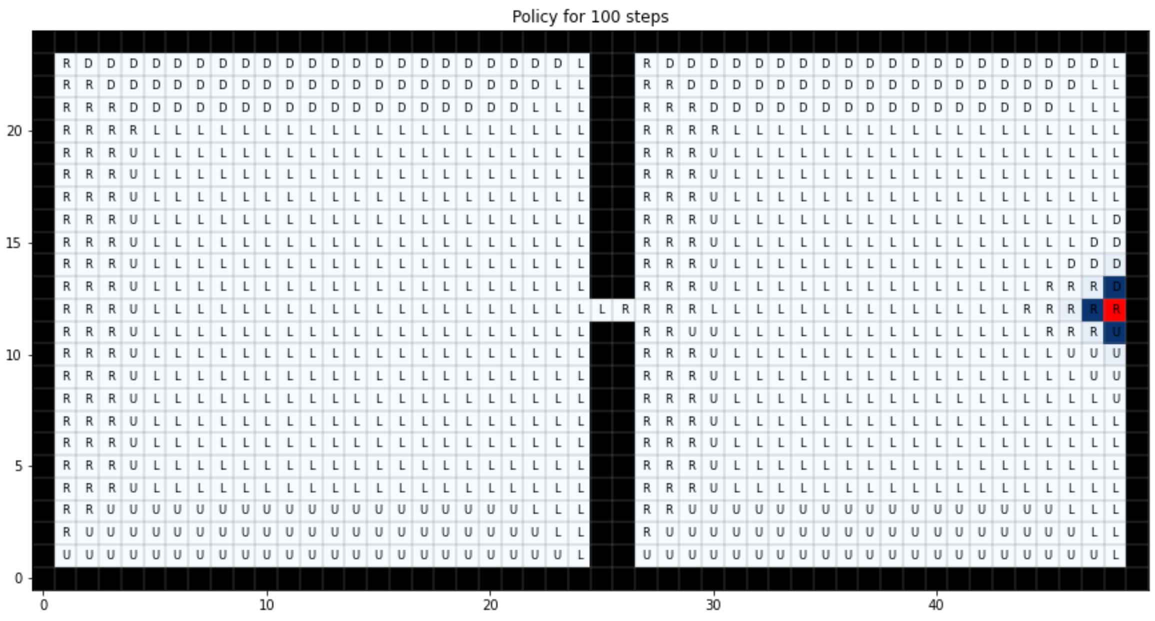
In this problem we had to solve an **MDP (Markov Decision Process)**. We had a robot in a grid world of dimensions 50 * 25. We were given the transition model as well as he reward model. The transition model was stochastic i.e the intended action happened with a certain probability. We also had a goal state which gave a very high reward. Solving the MDP meant that we had to get the final optimal policy for the robot. **Optimal policy is the one in which the future expected reward is maximized.**
In the plots the darker blue cells represent that the value there is higher. Also the goal state is represented by a **red color**. In the policy markings, L means Left, R means Right, U means Up and D means down. The black cells represent the walls.

1.2 Part a - Implementing value iteration

In this part we had to implement the value iteration algorithm. This is a combined name for policy evaluation and policy improvement. The parameters given were, discount factor $\gamma = 0.1$ and max norm $\theta = 0.1$. We were required to simulate for a maximum of 100 iterations. But, the values converged at step 4 for a max norm of 0.1. Observing the policy at this step, **it was far from optimal**. On simulating to a 100 steps, the policy obtained was much better than that obtained at 4 steps, but one can easily observe that the **policy is not optimal** and we can take some better actions for certain states. The reason behind the policy not being optimal even at 100 iterations, is that the discount factor, γ is too low and thus for the higher number iteration, the effect is too low to make any difference to the values or the policy.



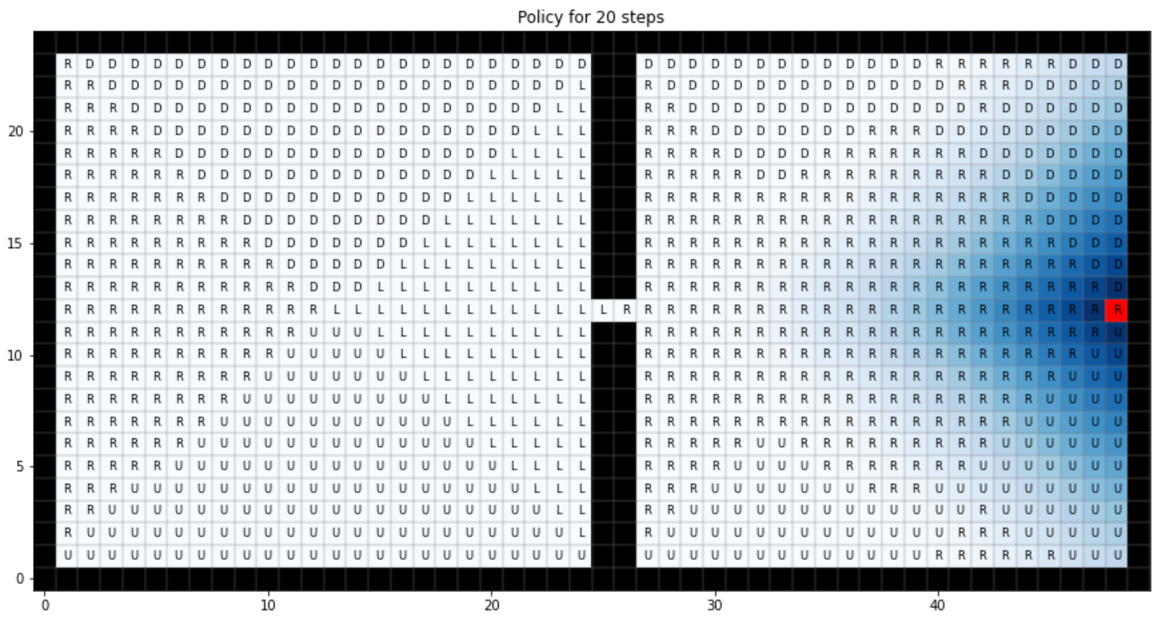
The above is the policy for when the value converges at step 4



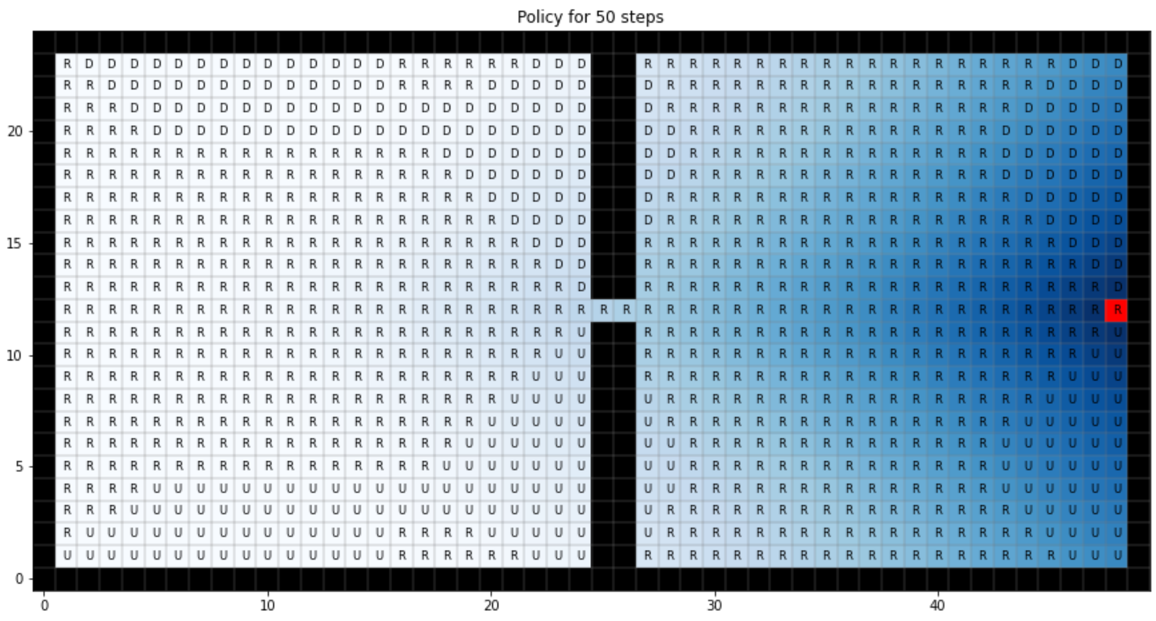
The above is the policy for 100 iterations of the value iteration algorithm

1.3 Part b - Value iteration for $\gamma = 0.99$

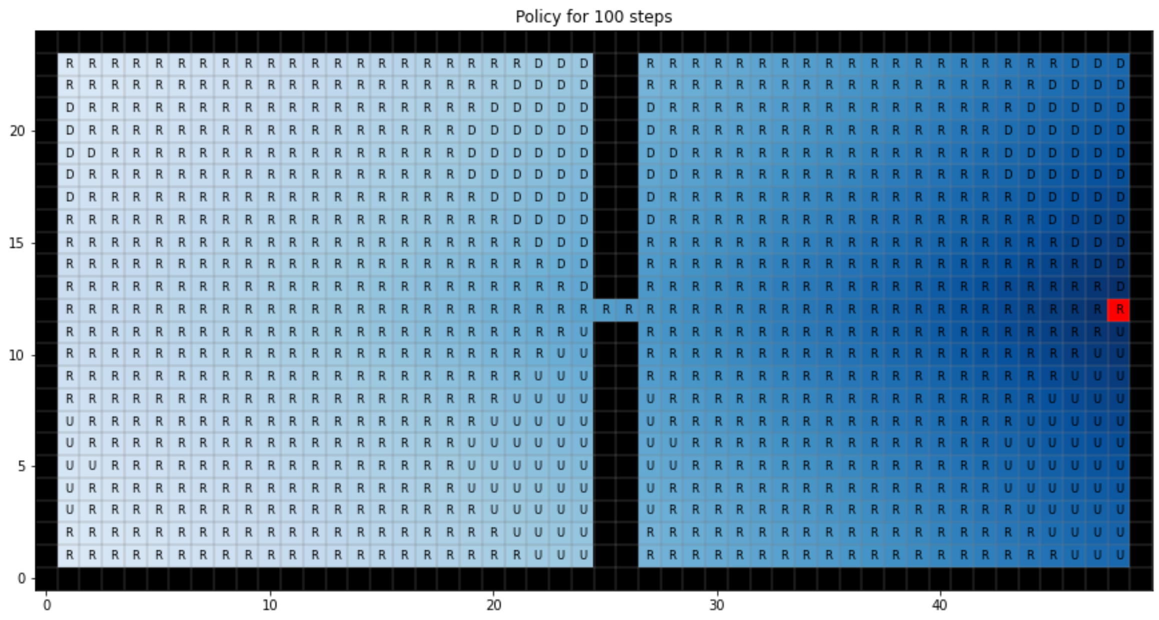
In this we had to do the same thing as part a but with one change, the discount factor is now 0.99. We can clearly see the differences between this and $\gamma = 0.1$. We can see that the value converges after much more iterations (**662 steps for value convergence**). The **final optimal policy that we get is optimal** and we get the best intended actions for every state. This is because in this case the discount factor is higher, and thus the even the higher numbered iterations are affecting the values and the policy. Also, as we will see in part d, the policy converges much much before the values converge.



The above is the policy and value plot for 20 iterations



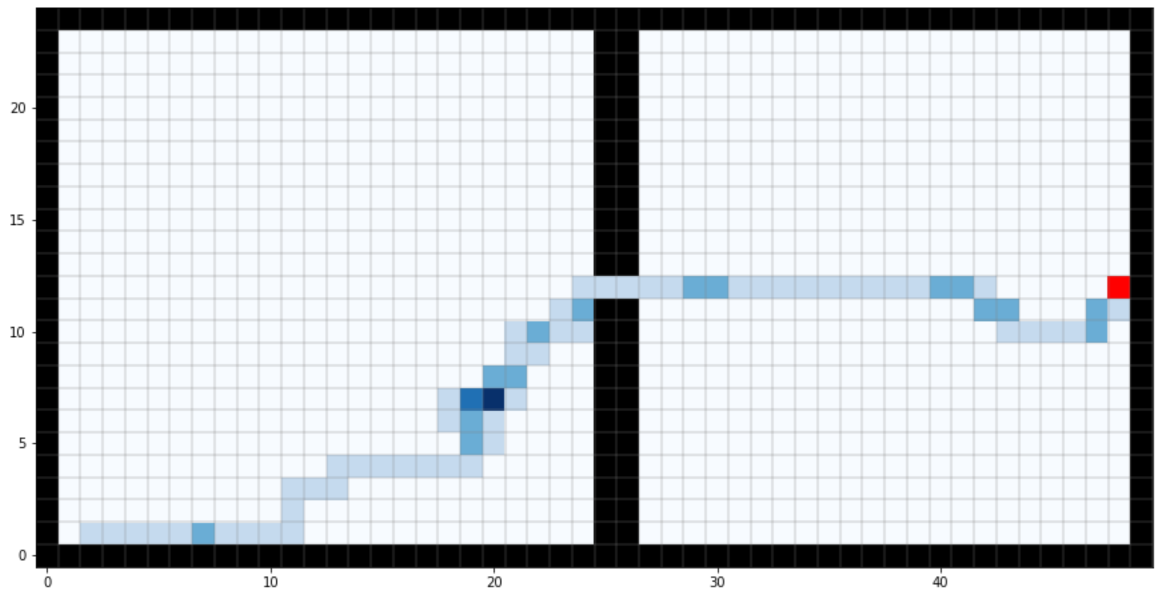
The above is the policy and value plot for 50 iterations

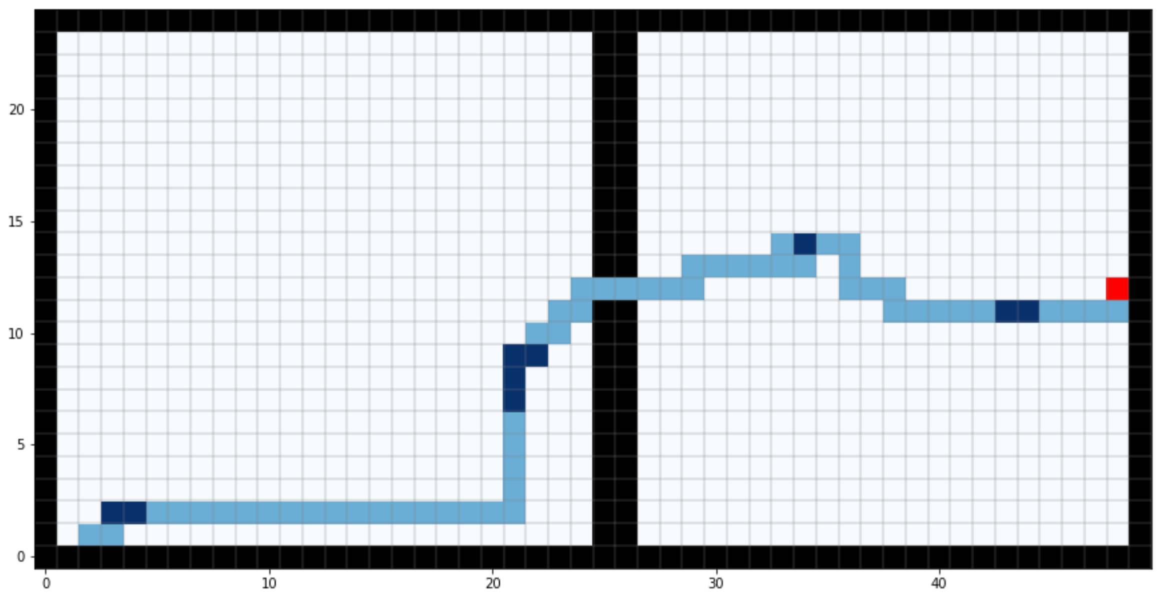


The above is the policy and value plot for 100 iterations

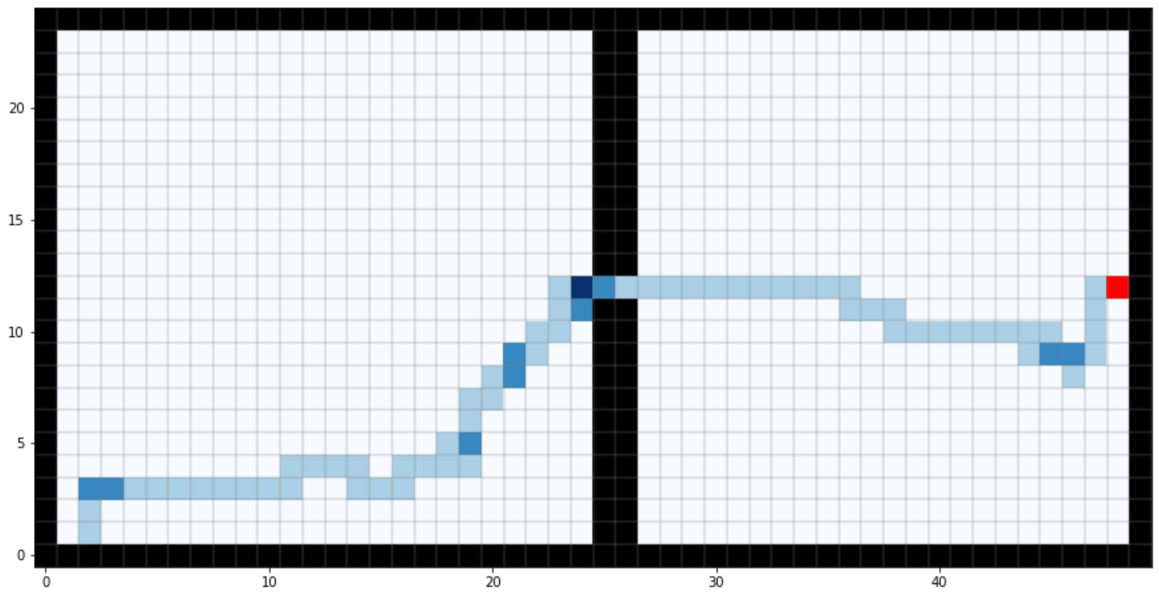
1.4 Part c - Sample execution of policy

In this part we had to execute the motion of the robot from an initial position of (1,1) using the final policy obtained for $\gamma = 0.99$. The motion would terminate when the robot reaches the goal state or when the robot has executed 1000 steps. This motion was simulated 200 times (or for 200 episodes). Also we can clearly see from the motion that the **robot quickly goes to the goal state** because the policy is optimal and the transition model has a relatively high probability of 0.8 to execute the intended action. Just so that the report doesn't become too long, I'm only attaching 10 such paths that I obtained.
Another observation that can be made through these plots is the **stochastic behaviour of the environment**. This is the behaviour that has been defined by the transition function. Even though the optimal policy is in place, the agent deviates from the optimal path because of this stochastic environment.

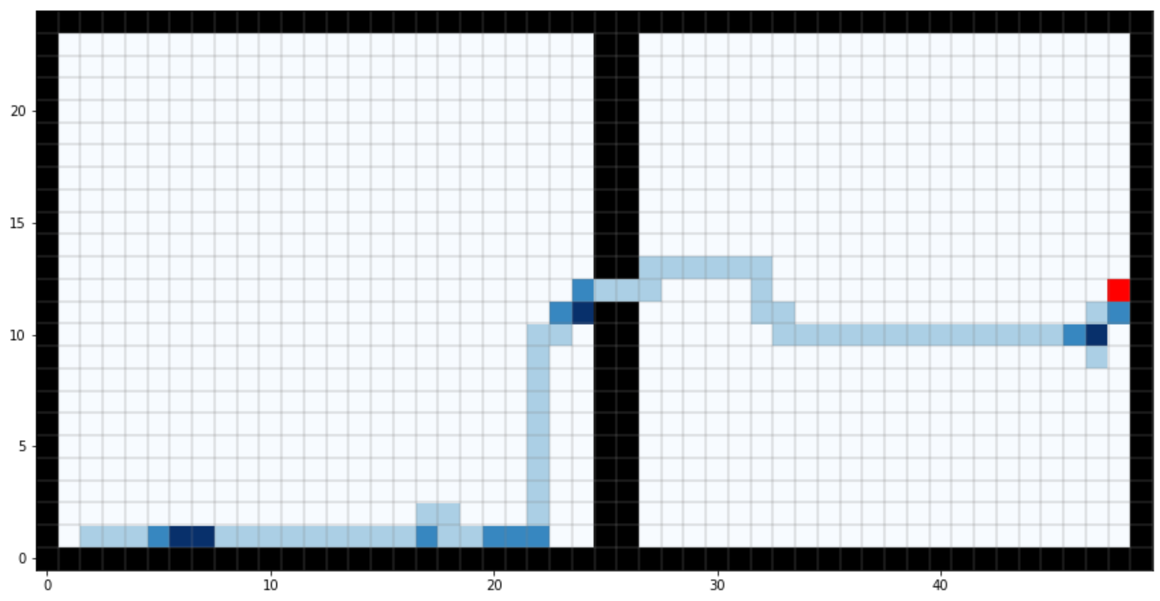


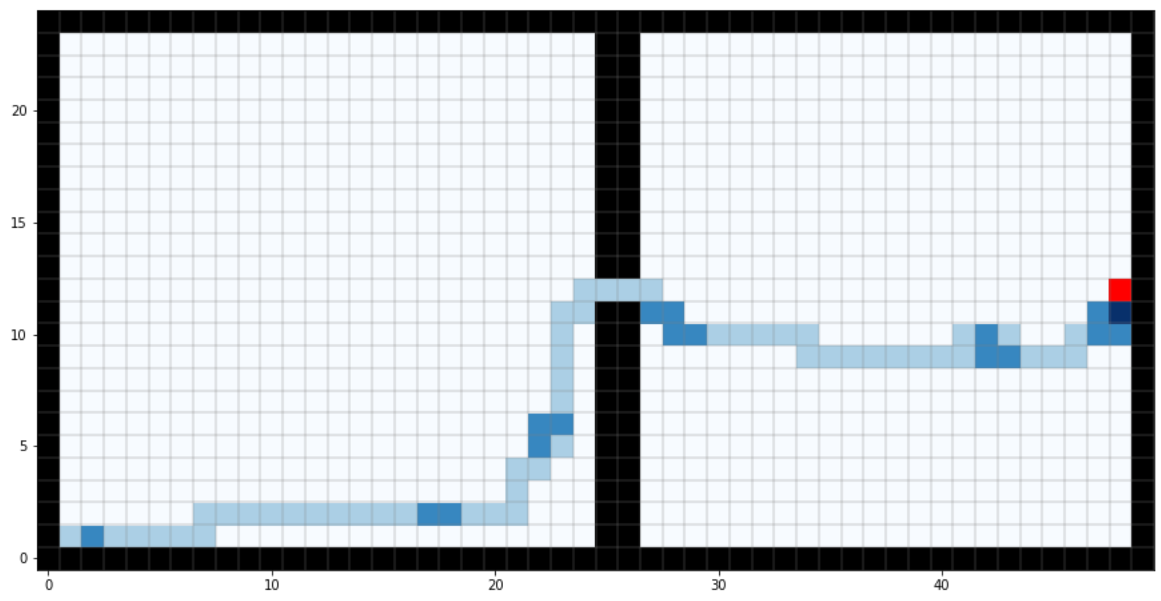


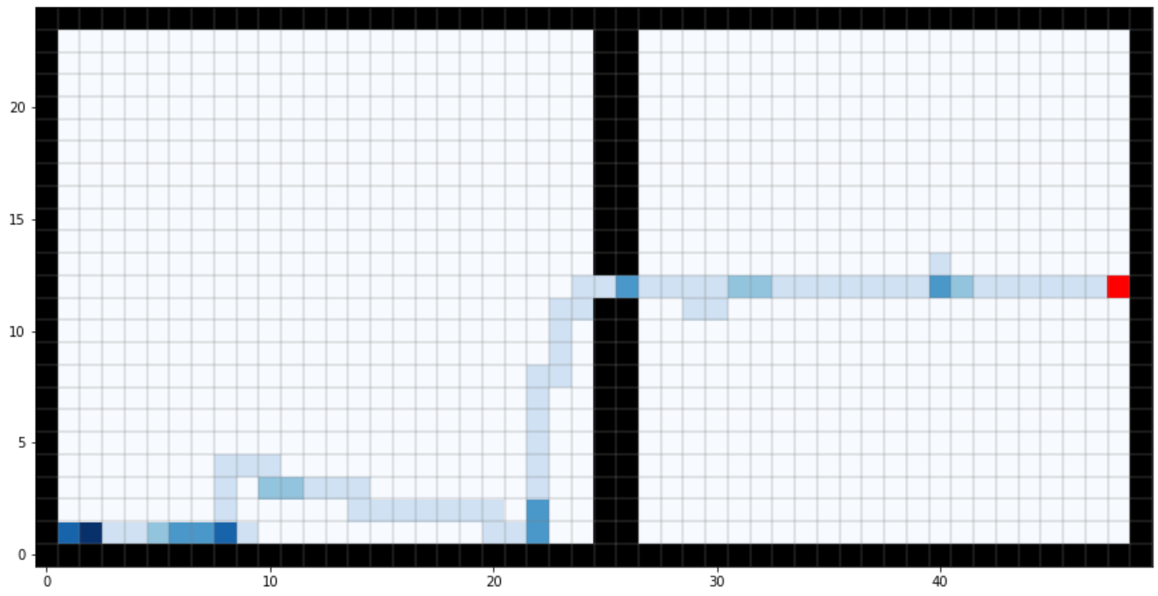
0
5
10
15
20
0
10
20
30
40

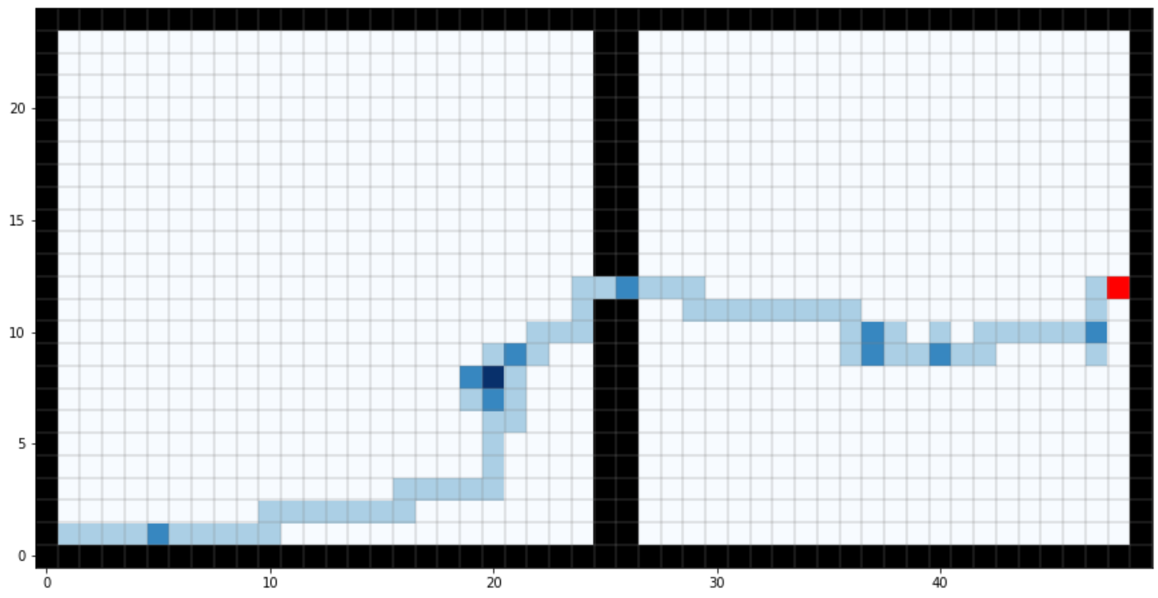


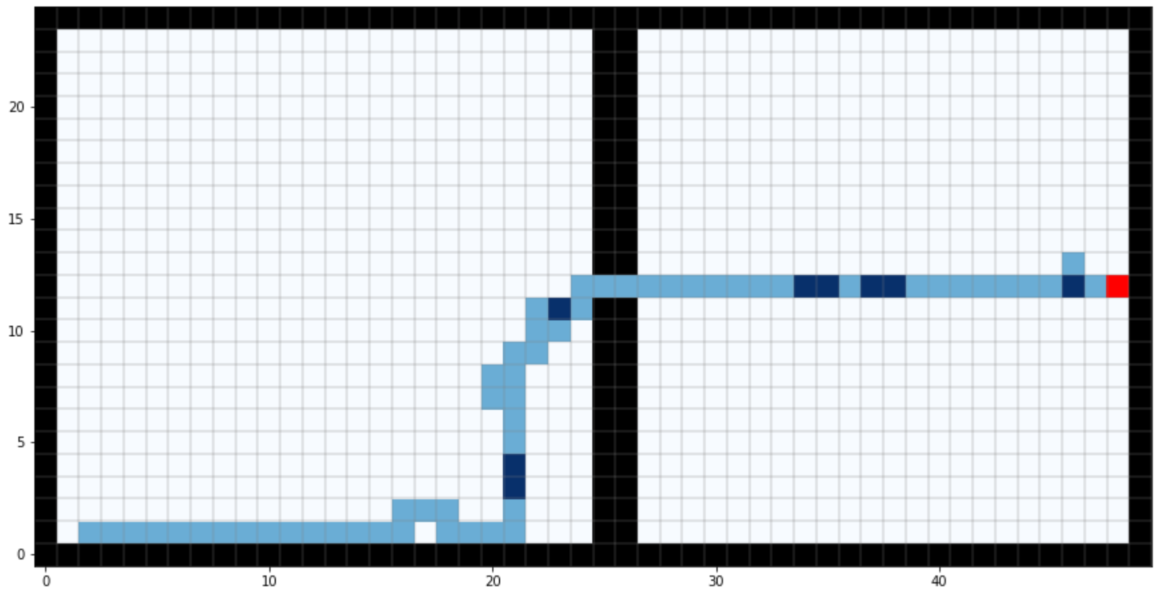
0
5
10
15
20
0
10
20
30
40


0
5
10
15
20
0
10
20
30
40

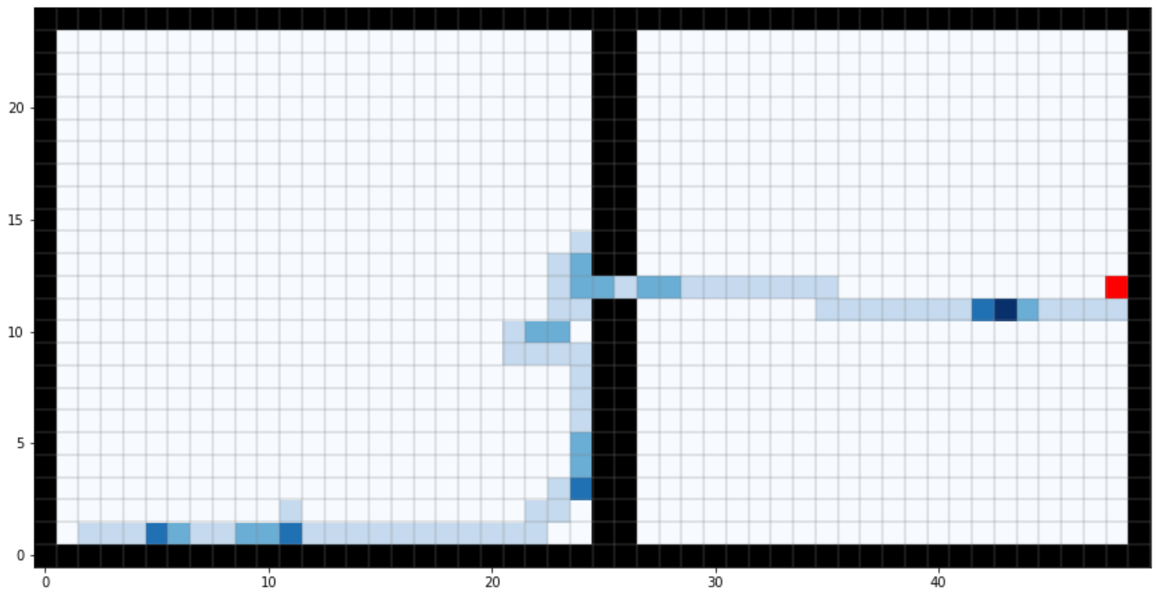

0
5
10
15
20
0
10
20
30
40


0
5
10
15
20
0
10
20
30
40

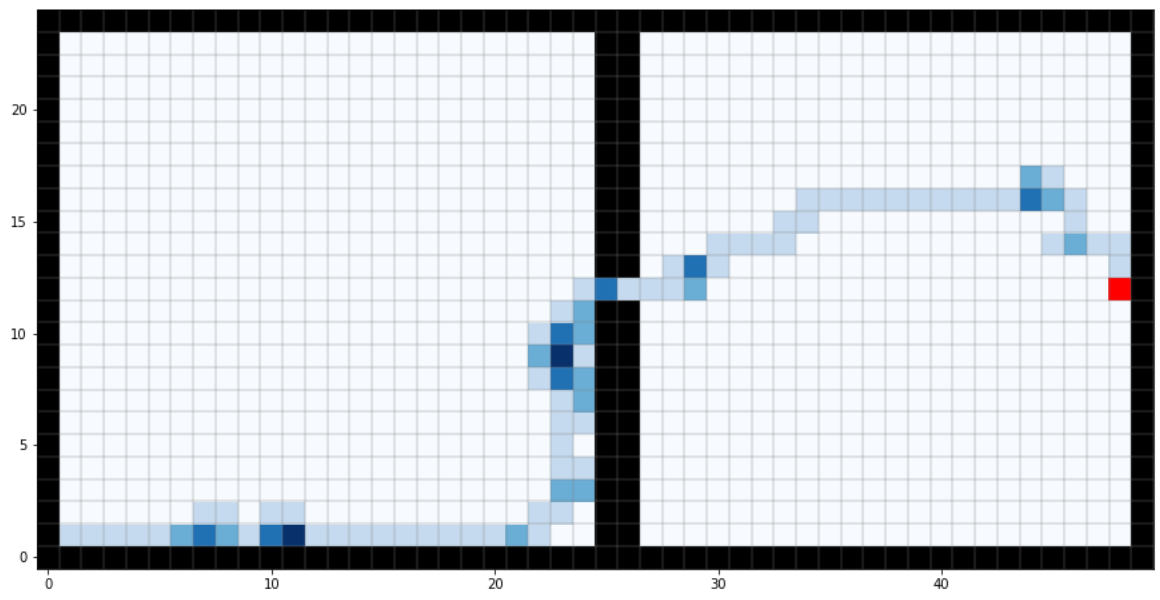

0
5
10
15
20
0
10
20
30
40



0
5
10
15
20
0
10
20
30
40

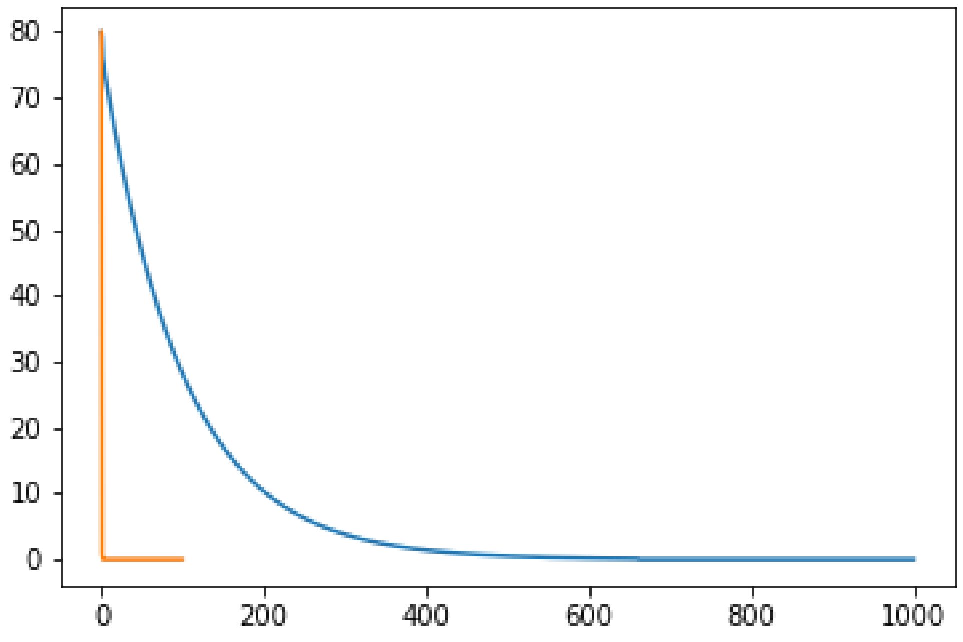


0
5
10
15
20
0
10
20
30
40



1.5 Part d - Max Norm Trend and Policy Convergence

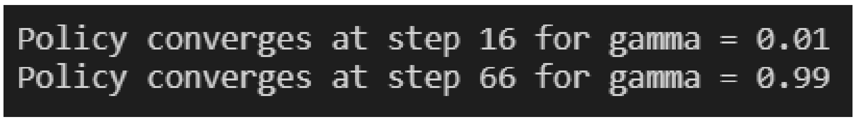
The graph for max norm vs iteration number is shown below. The blue corresponds to $\gamma = 0.99$ and the orange line corresponds to $\gamma = 0.01$.



Here, we can clearly see that the **max norm decreases with the increase in the iteration number**. This can be attributed to the discount factor because as the iteration number decreases the discount factor ensures that the effect of that iteration becomes lower and lower. This means that the change in values between successive iterations becomes less and less. This is exactly what max norm is. Also as the discount factor increases, the value function takes more and more time to converge.

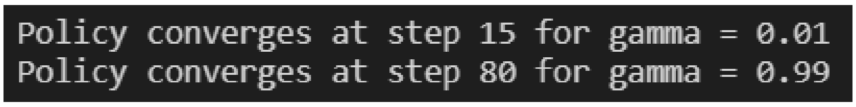
We also looked at policy convergence. I've considered two meanings of policy convergence:

- The policy is considered to be converged when the policy for two successive iterations comes out to be the same. Considering this definition, we got the following results:



```
Policy converges at step 16 for gamma = 0.01
Policy converges at step 66 for gamma = 0.99
```

- The policy is considered to be converged when the policy for a step becomes equal to the final policy i.e the policy at value convergence. For this, we got the following results:



```
Policy converges at step 15 for gamma = 0.01
Policy converges at step 80 for gamma = 0.99
```

As we can clearly see that in case of lower gamma i.e $\gamma = 0.01$, the policy converged after the value converges. Therefore, the policy obtained at value convergence is not optimal. But, in case of higher gamma i.e $\gamma = 0.99$, the policy converged way before the value converged. Therefore, **if we only wanted the policy we could have stopped at about 80 iterations instead of 662 iterations.**
From these results, we can conclude that there exists a delicate balance between the value of gamma and the relative convergence of value and policy. There must exist a gamma such that the values and policy converge at the same time.

2 Question 2 - Implementing Q - Learning

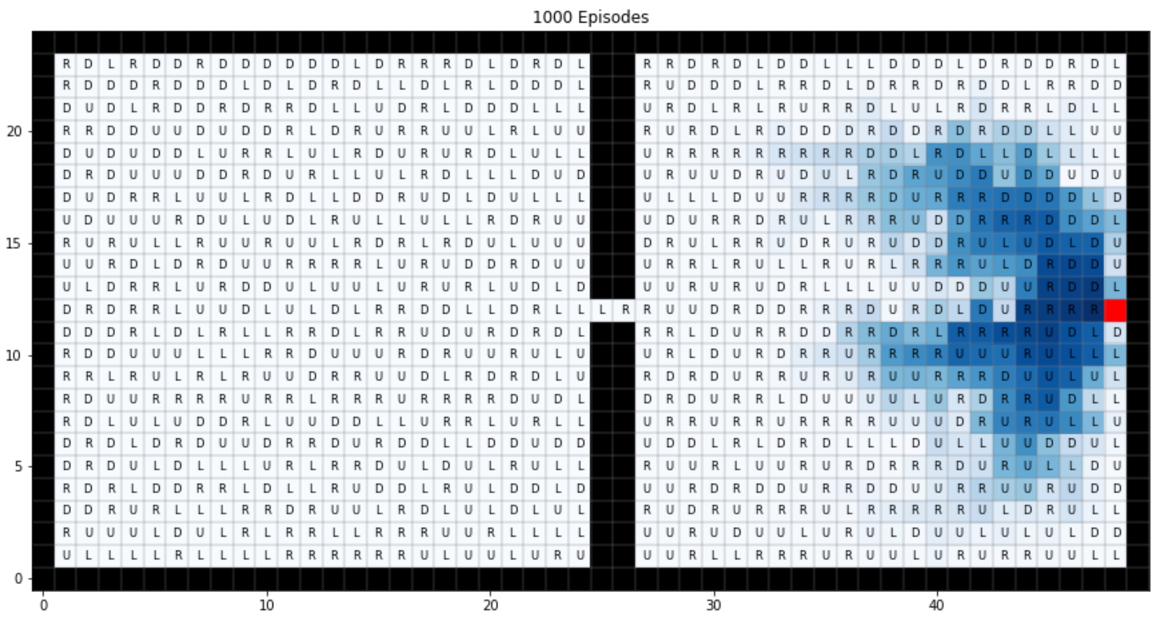
2.1 Overview

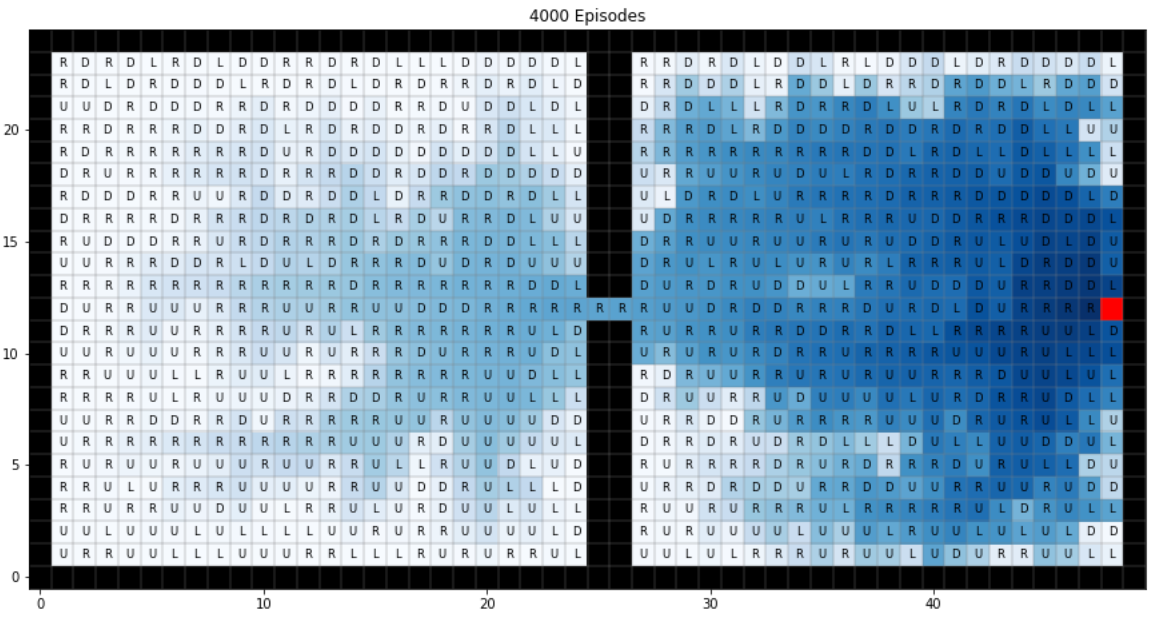
In this problem the **transition model and reward model are not at the agent's disposal**. The agent can however perform an action on a state and receive a reward from the environment as a feedback. This reward enables us to compute the Q - values for different states and on the basis of these q - values we can get the final policy for the agent. Q - Learning incorporates ϵ-greedy action selection. **This ensures that the robot explores all the states to pretty much the same extent.**

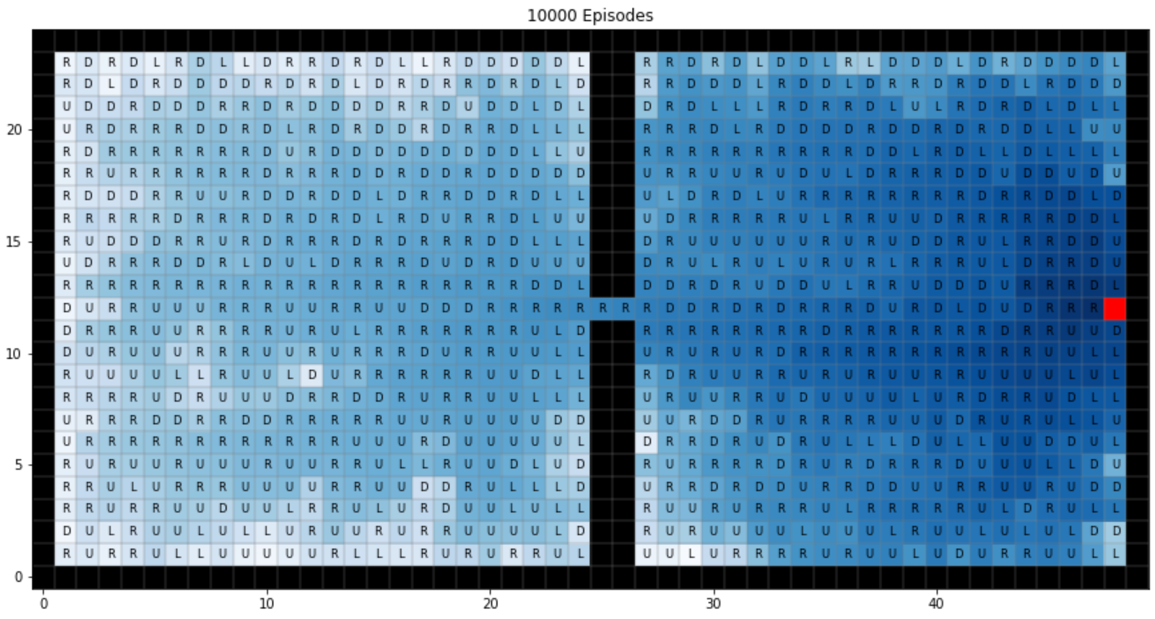
In the plots the darker blue cells represent that the value there is higher. Also the goal state is represented by a red color. In the policy markings, L means Left, R means Right, U means Up and D means down. The black cells represent the walls.

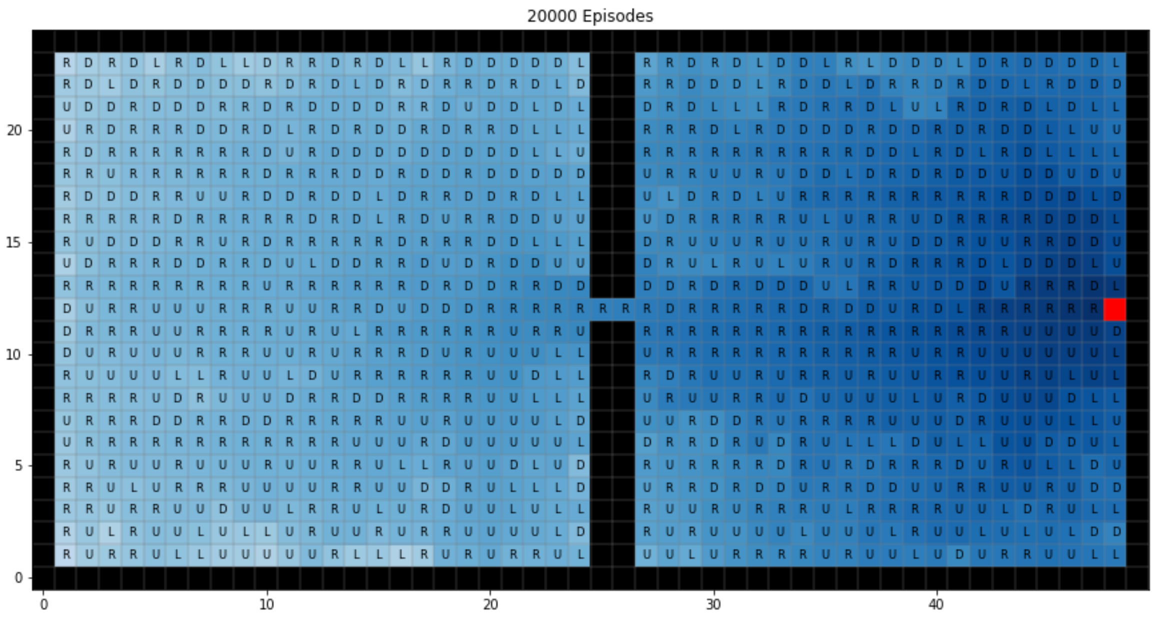
2.2 Part a, b - Q - Learning for $\epsilon = 0.05$ and plotting values and policy

For the first 2 parts of the problem we had to implement Q-Learning for $\epsilon = 0.05$ and thereafter plot the values and corresponding policy at different iterations. **I decided to go till 20000 iterations for better visualisation of the trend.** Following are the plots at 1000, 4000, 10000 and 20000 iterations respectively:


1000 Episodes
0
5
10
15
20
0
10
20
30
40


4000 Episodes
0
5
10
15
20
0
10
20
30
40

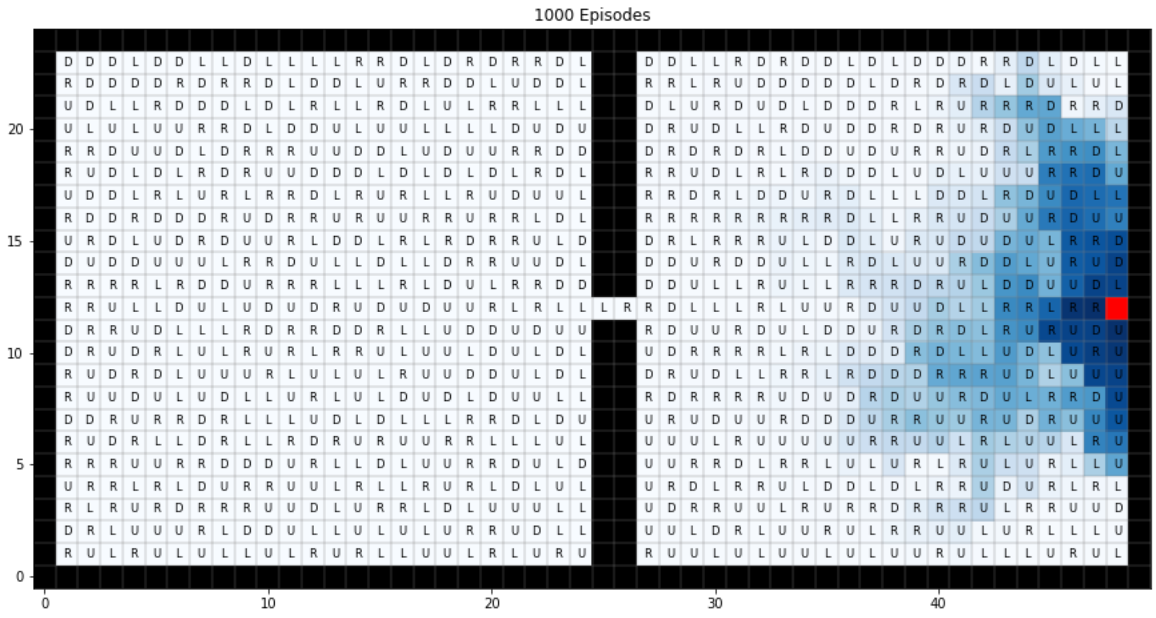


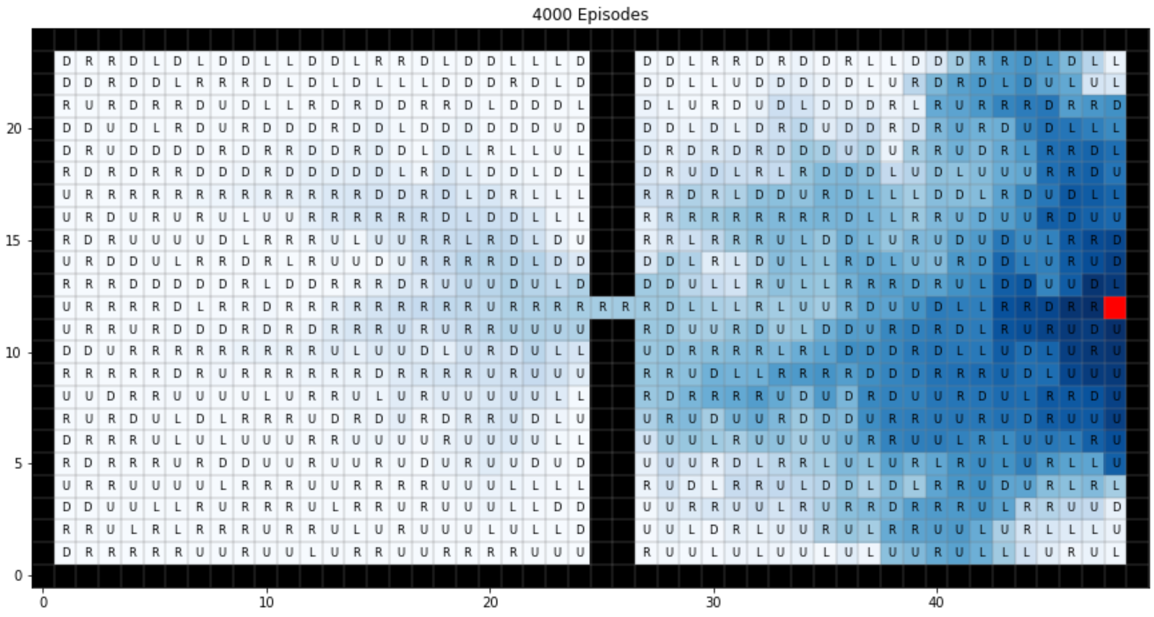


An important point to note here is that even though we go till 20000 iterations, **we do not get the optimal policy**. The policy does go to the goal state from every other state but it **does not take the shortest path**. This is a demerit of a lower ϵ or in other words lower risk taking.

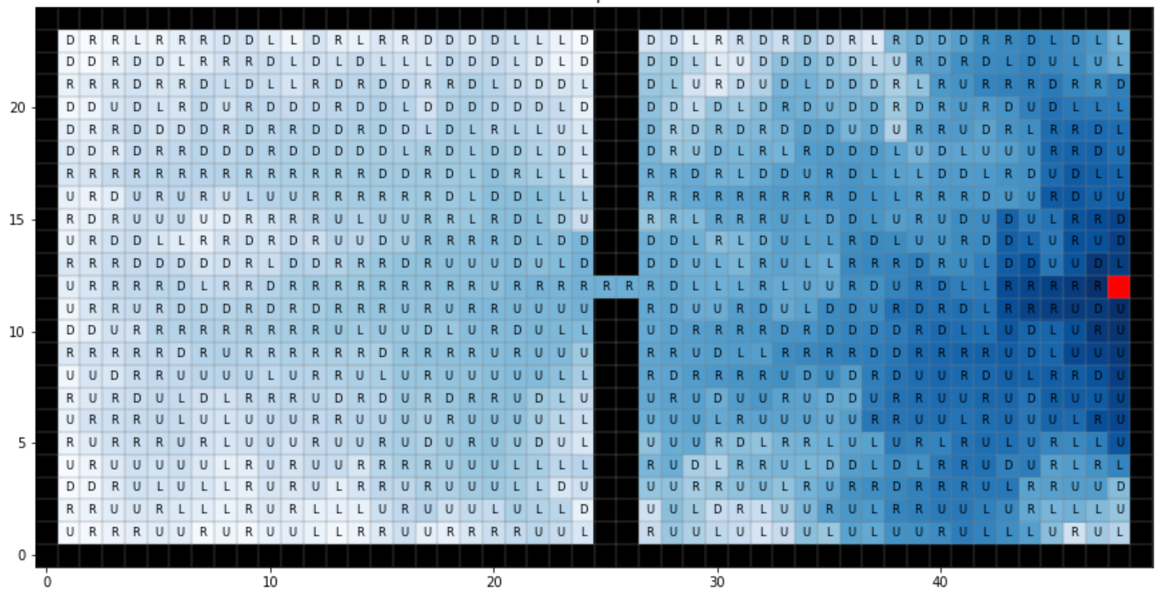
2.3 Part c - Q-Learning for $\epsilon = 0.005$ and $\epsilon = 0.5$

Following are the plots at 1000, 4000, 10000 and 20000 iterations for $\epsilon = 0.005$:

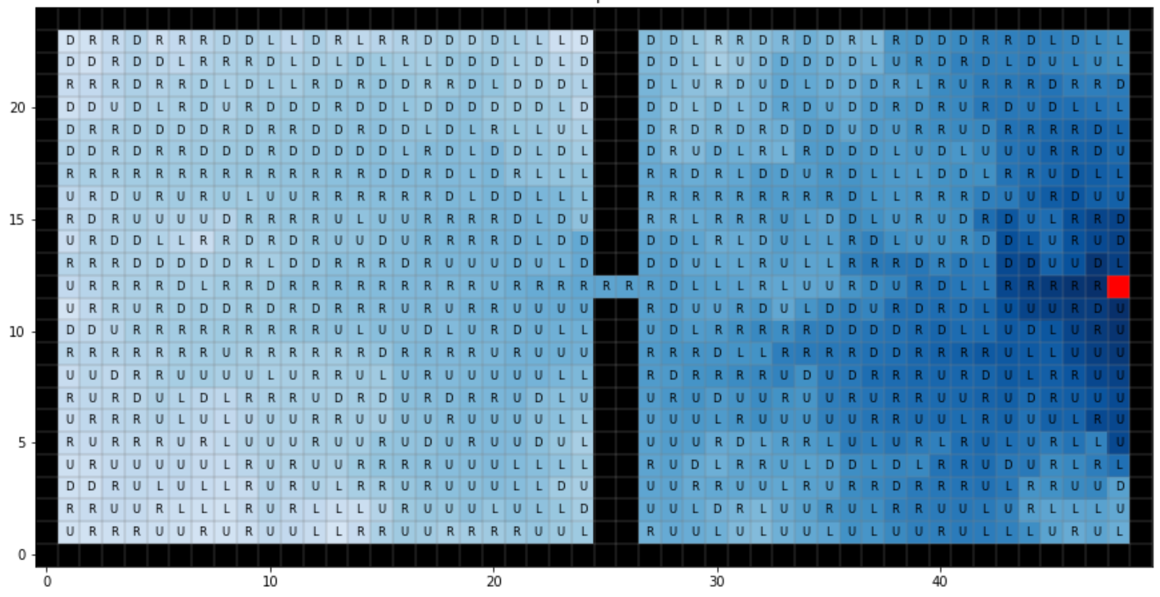




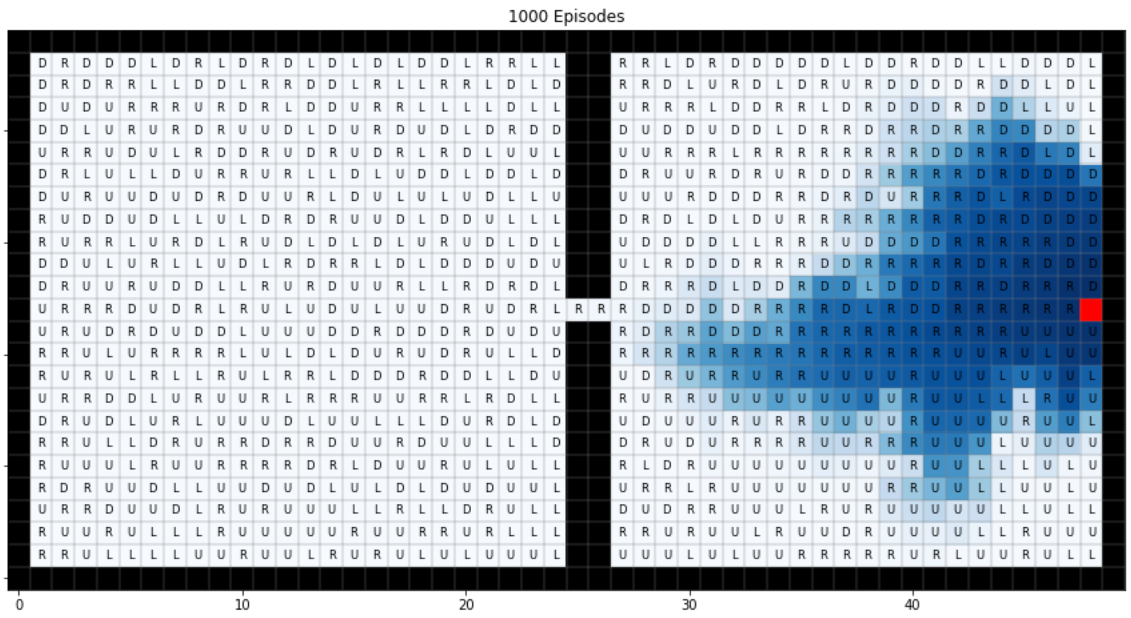
10000 Episodes

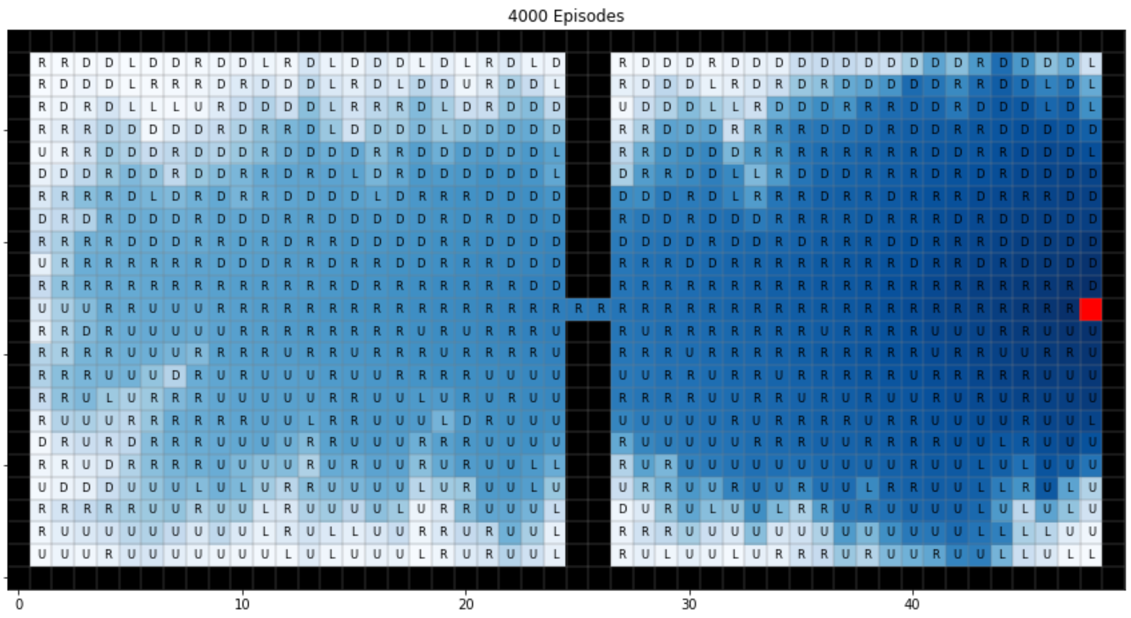


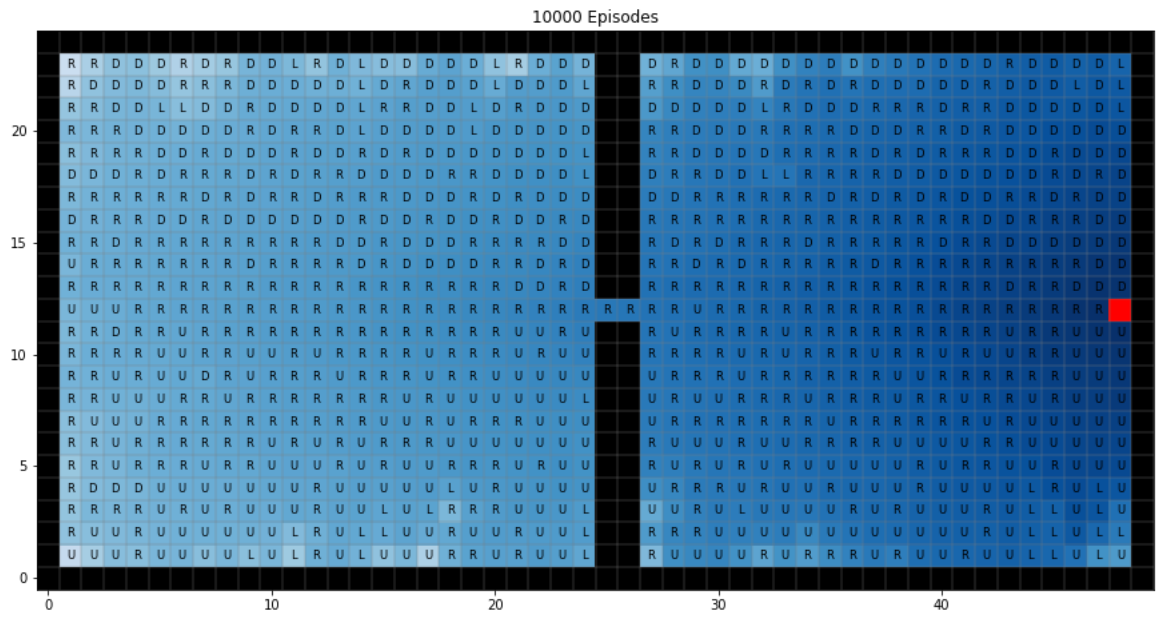
20000 Episodes

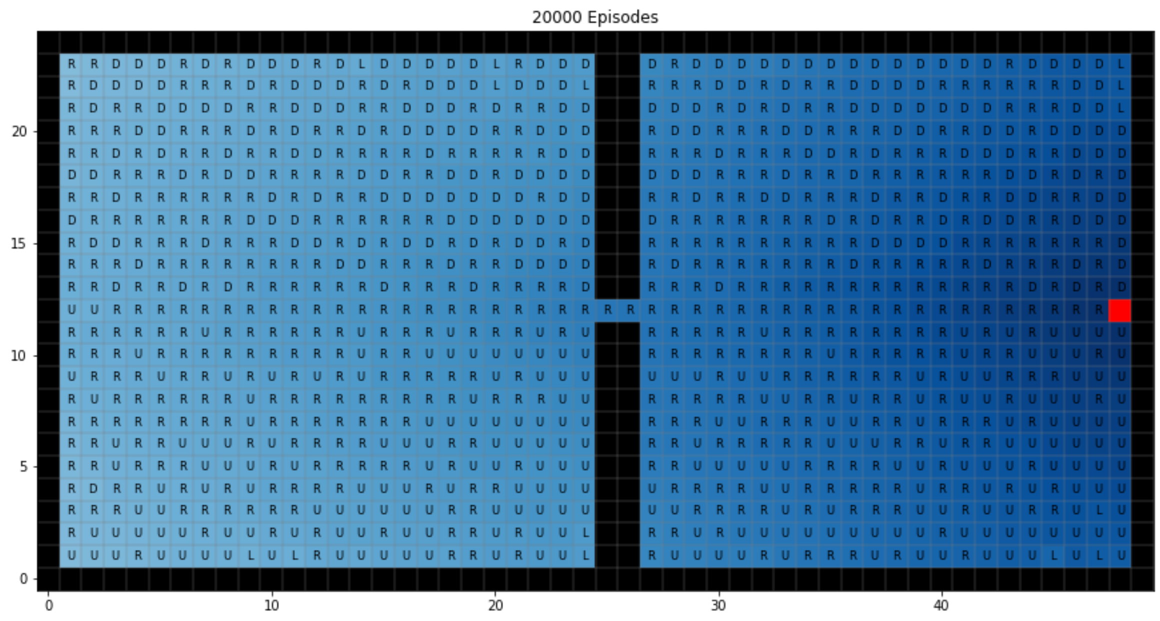


Following are the plots at 1000, 4000, 10000 and 20000 iterations for $\epsilon = 0.5$:







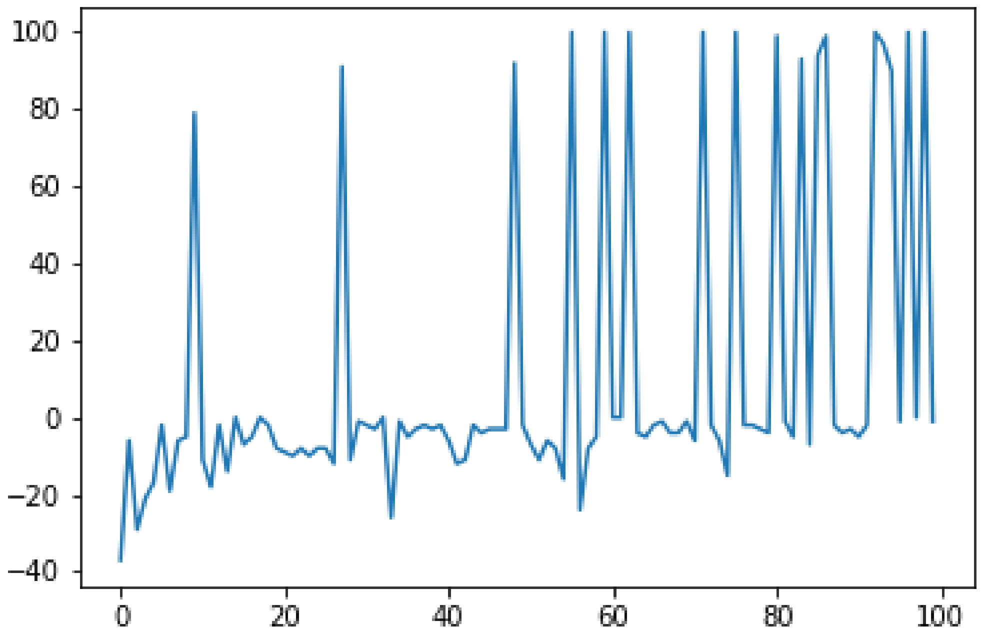


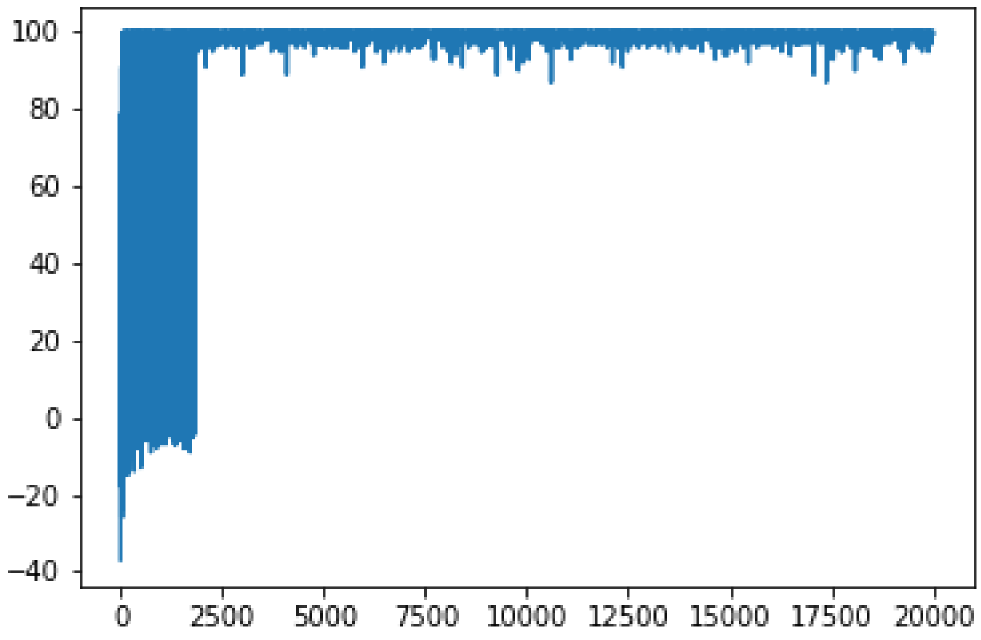
A few observations to make from these plots:

- We can clearly observe that as epsilon increases, **so does the spread of the plot at every iteration**. This is expected since greater ϵ means more risk taking capabilities and thus **better exploration**.
- Another observation that can be made is that as epsilon increases, we get a better policy at every iteration. For $\epsilon = 0.5$, the policy is **near to optimal at 20000 iterations** but for $\epsilon = 0.005$ and $\epsilon = 0.05$, its not that optimal.
  Higher ϵ means that there will be more exploration and we will explore further away from the initial state of the episode. There is a trade off between **exploitation and exploration.**

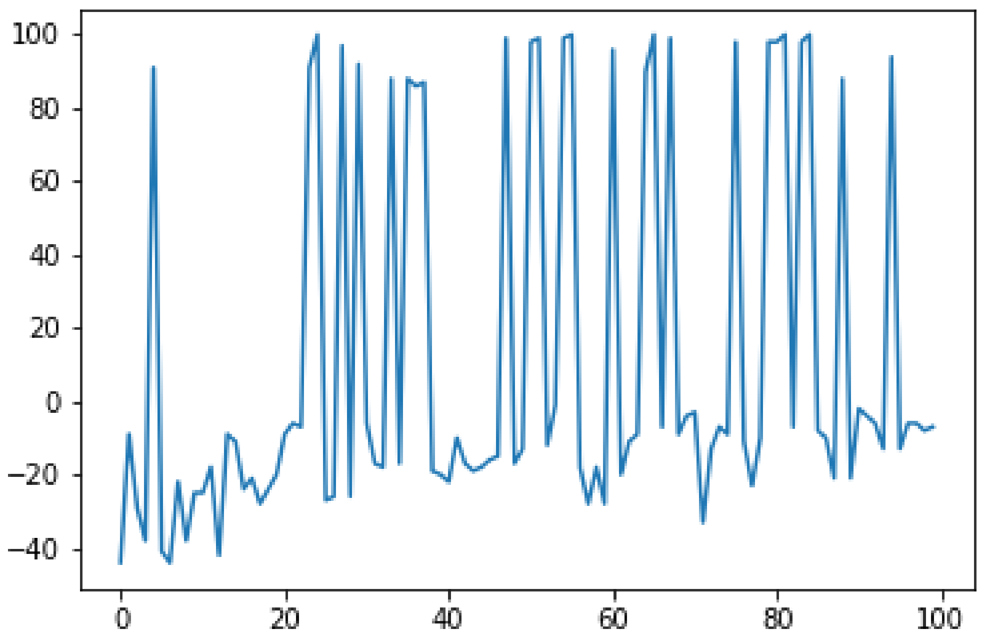
2.4 Part d - Rewards accumulated per episode

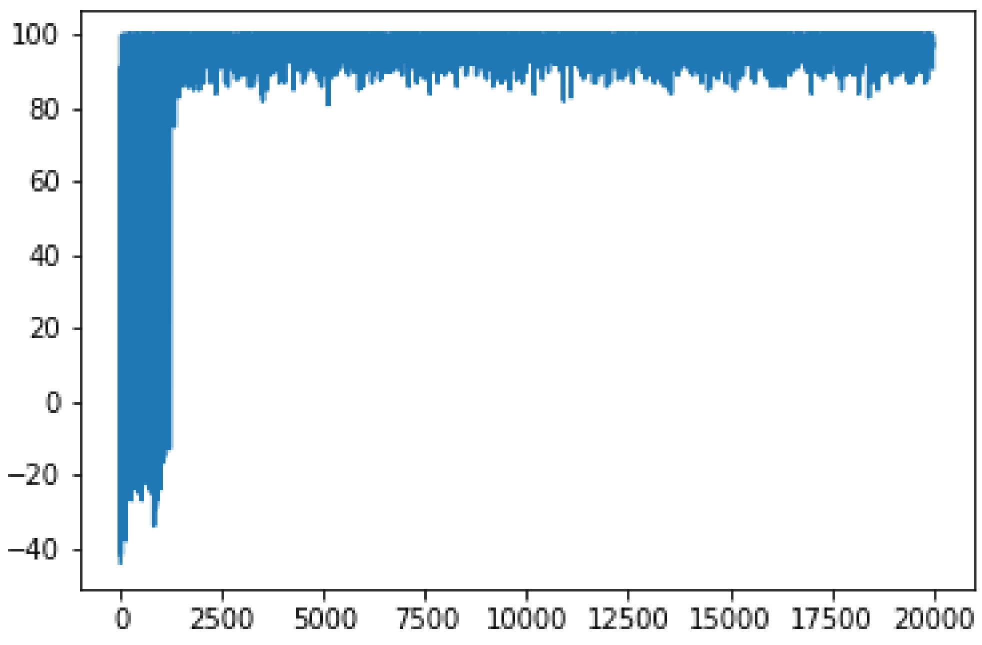
In this, I plotted the rewards accumulated per episode for 100 episodes and then for 20000 episodes. Here are the plots for $\epsilon = 0.05$:





And here are the plots for $\epsilon = 0.5$:





Three observations can be made from these plots:

- The important observation is that we can see that when $\epsilon = 0.5$, the rewards become consistently higher at around 1500 episodes. But, for $\epsilon = 0.05$, this comes around 2500 episodes. This is expected since higher epsilon explores much faster and therefore **knows about the goal state earlier than a lower epsilon.**
- The negative spikes in case of $\epsilon = 0.5$ i.e a higher epsilon go lower than the negative spikes for a lower ϵ. This behaviour is also expected **since more exploration results in more**

collisions with the wall and therefore a higher negative reward.

- In the plots up to a 100 episodes we can see that a higher ϵ reaches the goal state more often than a lower ϵ. This can be seen from the number of positive spikes. This behaviour is expected since **more exploration means reaching the goal state more often.**

3 References

- Artificial Intelligence: A Modern Approach by Peter Norvig and Stuart Russell
- Lecture slides