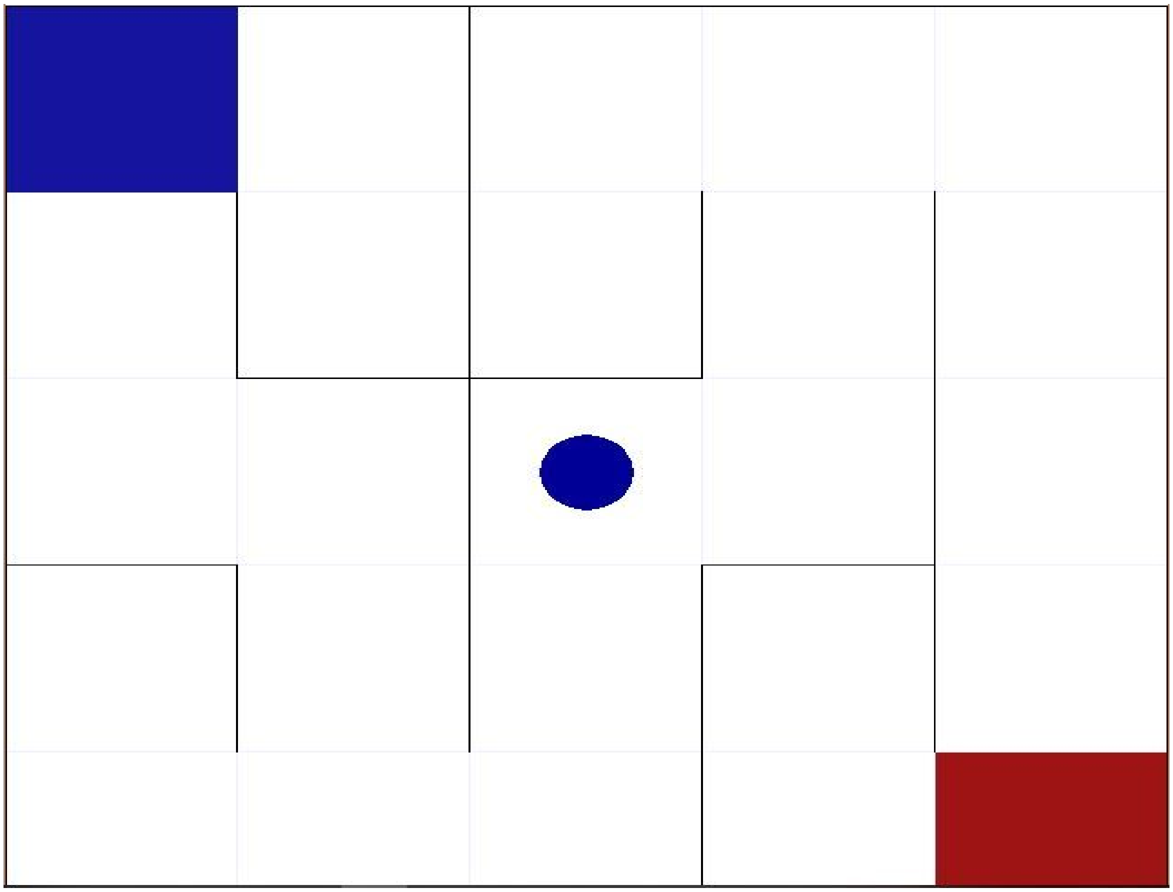
AI Assignment 4

Suraj Pandey - MT18025

3. a)
Q(st,at)=Q(st,at)+learning_factor*(reward_t+discount_factor*max_on_a(Q(st+1,at+1))-Q(st,at))
Training is done in different aspects i.e, using combination of different parameters value on same maze

Maze is:



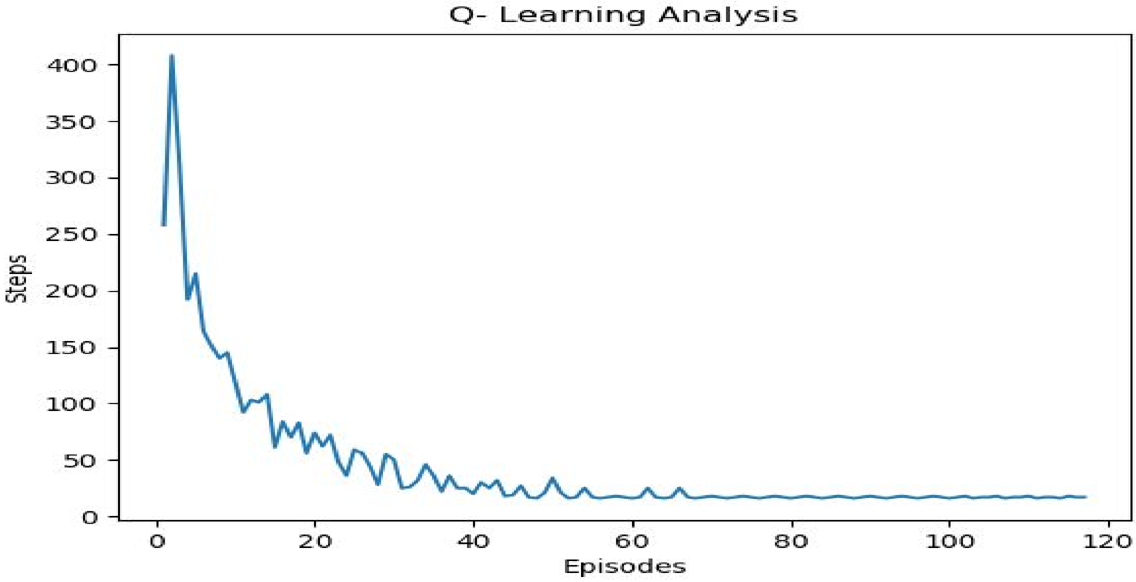
Different hyperparameters used to train the agent in maze to go to final goal.
These are:

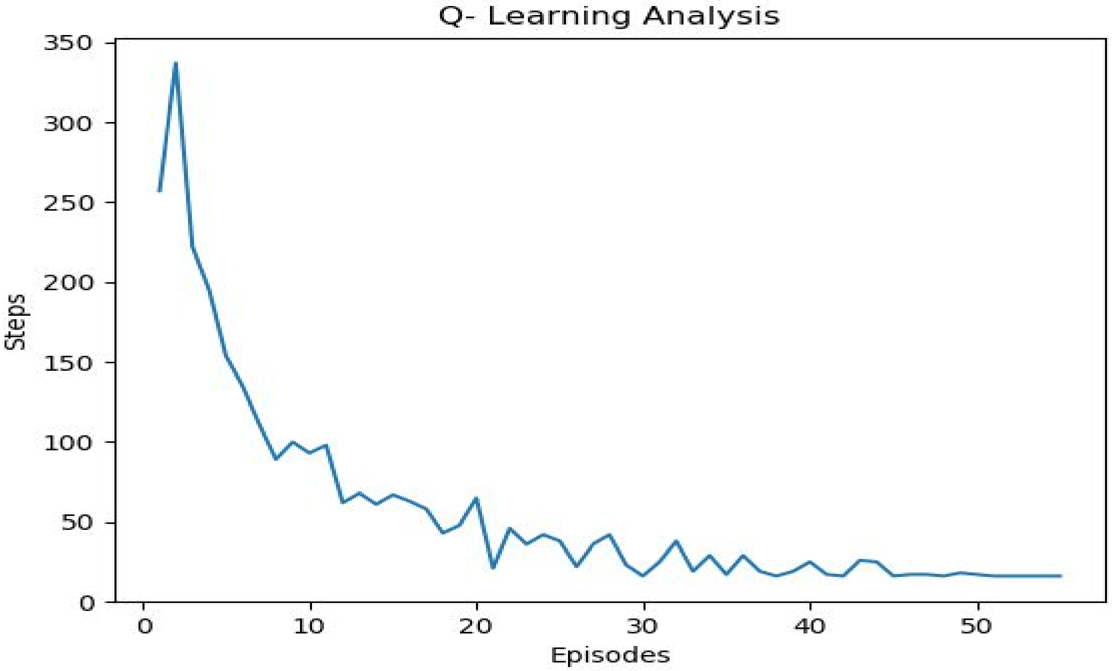
1. Learning rate
2. Discount Factor

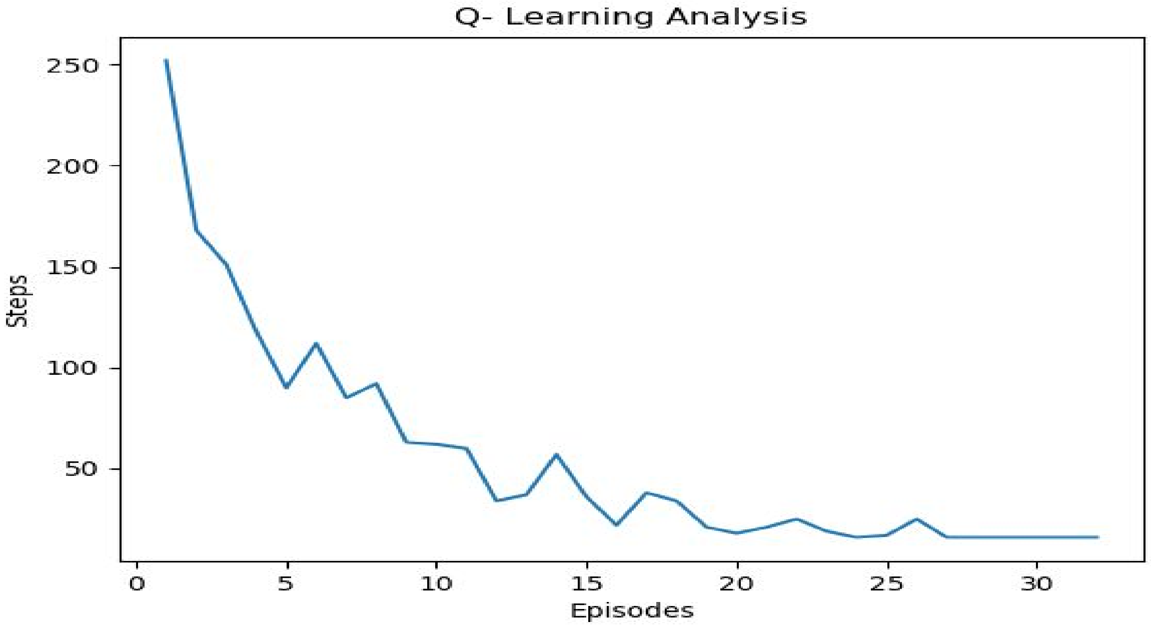
Result:

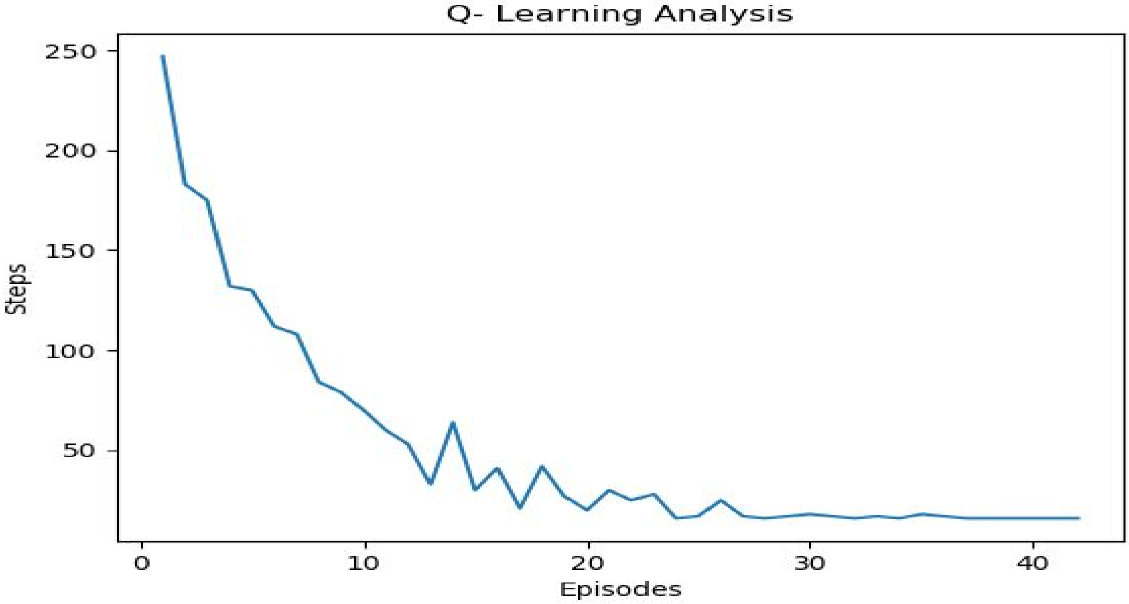
S. No.	**Learning Rate**	**Discount Factor**	**Episodes Taken**	**Time taken**
1	0.2	0.4	117	103
2	0.2	0.2	253	169
3	0.2	0.8	55	62
4	0.4	0.8	32	38
5	0.4	0.6	42	42
6	0.6	0.8	25	25
7	0.8	0.6	23	23
8	0.8	0.8	22	39

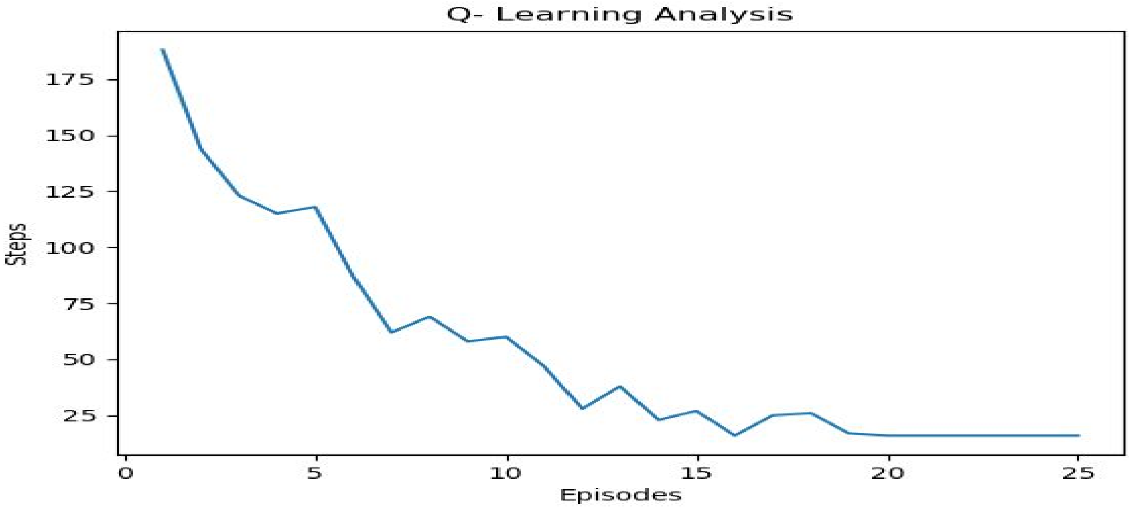
Plots are serially as according to Table.

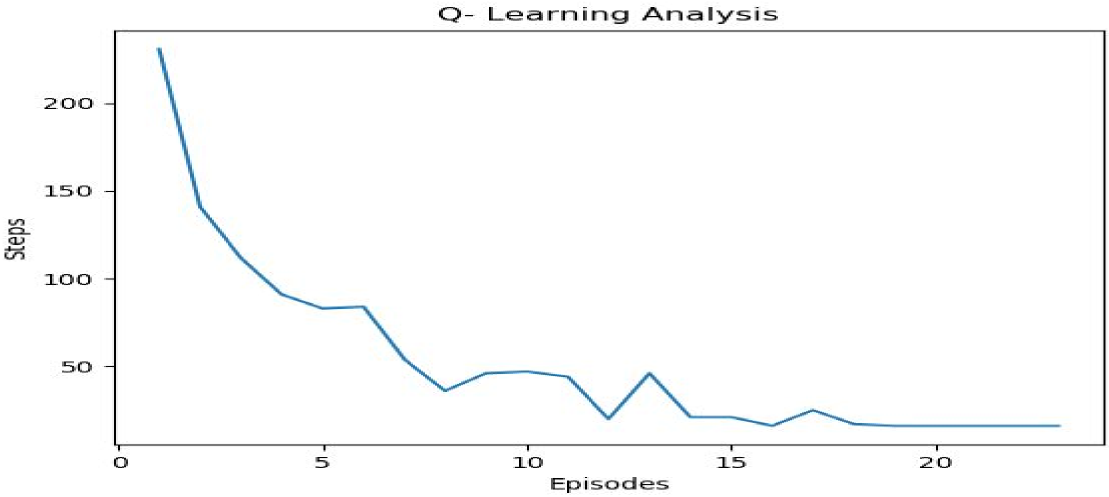


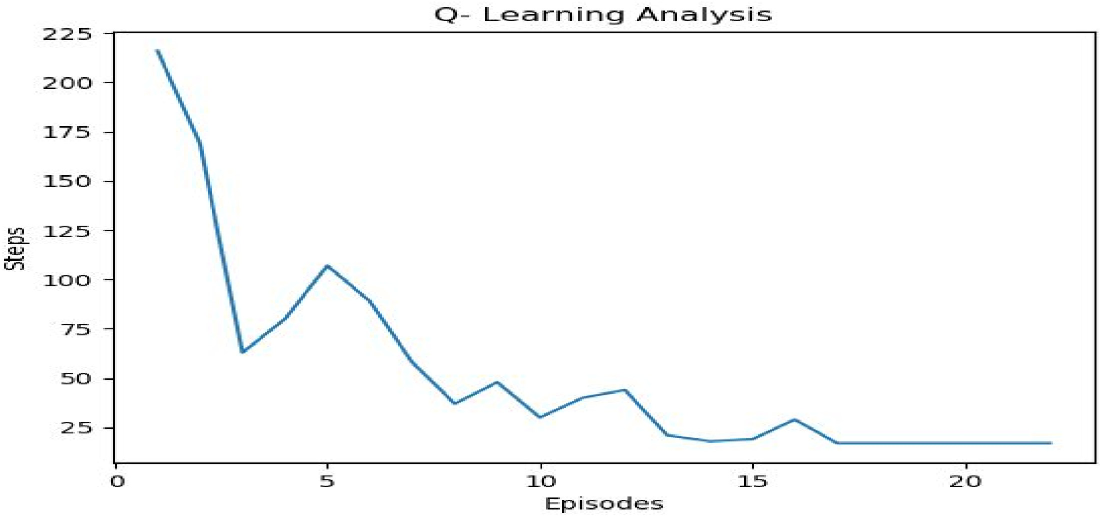

Q- Learning Analysis
Steps
350
300
250
200
150
100
50
0
0
10
20
30
40
50
Episodes


Q- Learning Analysis
Steps
250
200
150
100
50
0
5
10
15
20
25
30
Episodes


Q- Learning Analysis
Steps
250
200
150
100
50
0
10
20
30
40
Episodes


Q- Learning Analysis
Steps
175
150
125
100
75
50
25
0
5
10
15
20
25
Episodes


Q- Learning Analysis
Steps
200
150
100
50
0
5
10
15
20
Episodes


Q- Learning Analysis
Steps
225
200
175
150
125
100
75
50
25
0
5
10
15
20
Episodes

3.b)

The Agent is learning from the environment by rewards and value at each and every state.

By changing the hyperparameters it is behaving differently.

Hyper parameters are :

1. Learning rate
2. Discount Factor

Observations and Inferences:

The Convergence criteria dependent on the learning rate ,as the learning rate increases the steps taken to converges decreases.

So, the learning rate will affect the number of steps to converge.But the difficulty is :if the learning rate is large then, maze will not learn efficiently.

By the discount factor, we give weightage to the future rewards as in value of the state.

- If the discount factor is more then, it means giving the more weightage to the future rewards.
- And if it is less then, it means giving less weightage to future rewards.
- By increasing the learning rate and discount factor, the number of episodes taken to converge will be less but the time taken to complete one episode will be more because the weightage to future rewards become more and so, it will trace more paths as for learning.
- When learning rate is small, it will take more number of episodes because the learning speed is small and so it will learn slowly and so learn through may be more states than usual .
- When discount factor is less then, it will take more number of episodes to converge, because the weightage of the future reward is less and only dependent on present state‘ s reward and so take more number of steps to converge or learn.

3.c)

State Behaviour change towards the action :

Initially the agent is at start i.e, the initial position, then all actions is of same probability (uniform) as per Q-table has also values 0.

As per the iterations in episodes, the value in Q-table get changes and get higher value which has greater chance to take action.

The higher number in row of Q-table corresponds the action to perform for that state.

In other words,

Initially, for all i belongs to actions

p(action for particular state)=uniform

As per episodes:

p(action for particular state) changes as according to action performed in each episode.

Q-Table Value after training the maze by the agent:

```
[[-0.0066210910079018154, -0.00661636714147668, -0.006498203574970651,
-0.0066210910079018154], [-0.006626727550515397, -0.0066393896007551495,
-0.006639335155576441, -0.006639335155576441], [-0.00664092884138266,
-0.0066393896007551495, -0.006639335155576441, -0.006639335155576441],
[-0.0066413197264958675, -0.00663937190607207, -0.006639335155576441,
-0.006639335155576441], [-0.006641712915988952, -0.006639335155576441,
-0.006639328349929104, -0.006639335155576441], [-0.006555146284315288,
-0.006554474380034223, -0.0060856788454847445, -0.006557433907399097],
[-0.006463646713589307, -0.006455346527025306, -0.006424239280744891,
-0.006455346527025306], [-0.004809993493270653, 0.005598090384796984,
-0.004817460445797728, -0.004809993493270653], [-0.0038021733569580237,
-0.0035706394214400005, 0.03819591799098862, -0.0035706394214400005],
[-0.006639335155576441, -0.006639335155576441, -0.006639335155576441,
-0.006639339218511301], [-0.006426530144346939, -0.006426530144346939,
-0.005119647461836924, -0.00643399442919702], [-0.006211631610655955,
-0.005931982306196383, -0.006211631610655955, -0.006181419332929828],
[-0.005784182321998611, -0.00568684097399925, -0.005683322536790367,
-0.003691425194156813], [-0.0021235200000000003, 0.13468166600079864,
-0.0021235200000000003, -0.002335479808], [-0.0008640000000000001,
-0.0015040000000000001, 0.38512103969798894, -0.0008],
[-0.006149581375745403, -0.002660419870529771, -0.006149581375745403,
-0.006182925281985885], [-0.005625354292515464, -0.00568684097399925,
0.0035197491391561137, -0.00568684097399925], [-0.00568684097399925,
```

-0.00568684097399925, -0.00568684097399925, -0.005689303880045468], [-0.0008, 0.3949734467793621, -0.0008, -0.0008], [-0.0008, -0.0008, 0.9974266761884933, 0], [-0.004556810787807 56, -0.004787123566942344, -0.004556810787807 56, -0.004556810787807 56], [-0.004664184970207559, 0.018971053911490364, -0.004809993493270653, -0.004820890254 66779], [-0.003274416701 4400003, 0.05756223509962209, -0.0035706394214400005, -0.0035706394214400005], [-0.0021730816, -0.0021235200000000003, -0.0021235200000000003, 0.15398072067907403], [-0.006461578580757831, -0.006483554034363282, -0.006442584032537864, -0.006468744093154887]]

3.d)

By modifying the Q-Learning algorithm by adding the exploration rate in it, So the algorithm now have the probability to choose random move also except the Q-table value.

By this, the exploration part will be there in algorithm and so new path can be trace in learning of maze to explore it and see the expected sum from there .
If it is good path then , agent will learn that way otherwise not to take that way .

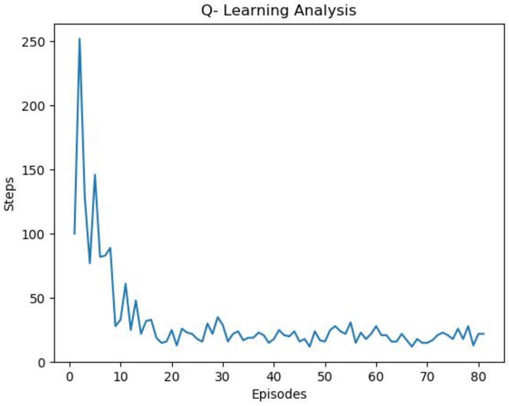
The exploration rate is in between o and 1.
By varying the parameter (Exploration rate) in maze ,results are:

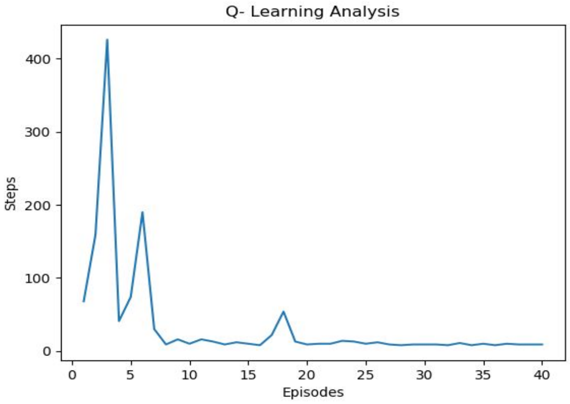
Results:

S.No.	**Learning rate**	**Discount Factor**	**Exploration rate**	**Episodes**	**Time taken**
1	0.4	0.3	0.4	81	46
2	0.4	0.3	0.2	53	32
3	0.4	0.5	0.2	90	66
4	0.2	0.2	0.2	78	47
5	0.6	0.4	0.1	54	26
6	0.8	0.5	0.2	35	22
7	0.2	0.4	0.6	218	78
8	0.2	0.8	0.4	71	32
9	0.2	0.8	0.8	87	102
10	0.2	0.4	0.01	54	46

Observations and Inferences (Effects):

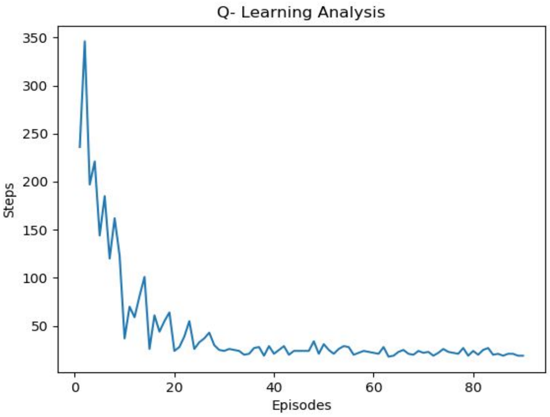
1. When the exploration rate is more (>=0.5) then, probability to select the random moves are more and so the learning will take the time and so the more number of episodes.
2. When the exploration rate is less (<=0.5) then, probability to select the random moves will be less and so learning will be done properly and in less time a.
3. Exploration looks important with less value when there is exploration to know the new state randomly without the Q-value.
4. Exploration is done to explore the new unseen state , uniformly among the action space in environment.
5. Exploration will take more number of episodes to converge or learn the maze.
6. More Exploration rate , more episodes taken to converge
7. Less Exploration rate, less episodes taken to converge.



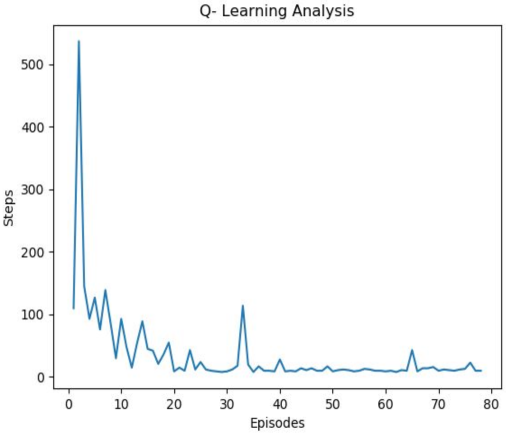


Learning rate:0.4
Exploration rate:0.4
Discount factor:0.3

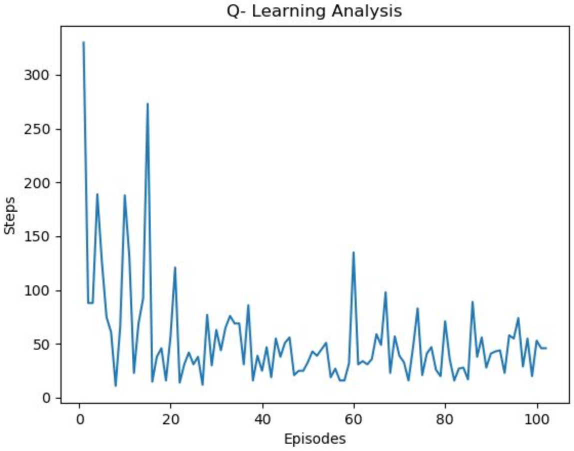
Learning rate:0.4
Exploration rate:0.4
Discount factor:0.2



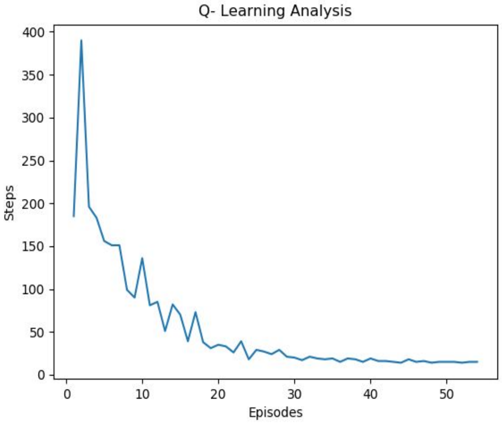
Learning rate:0.4
Exploration rate:0.5
Discount factor:0.2



Learning rate:0.2
Exploration rate:0.2
Discount factor:0.2



Learning rate:0.2
Exploration rate:0.8
Discount factor:0.8

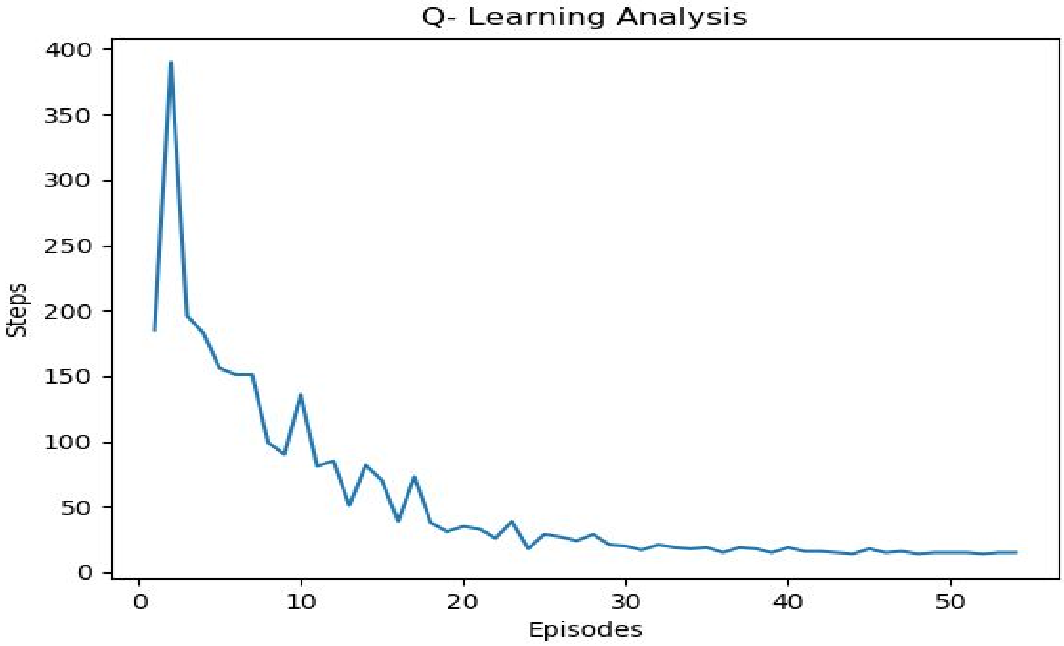


Learning rate:0.2
Exploration rate:0.4
Discount factor:0.01

3.e)

So, the best parameters are :

1. Less learning rate to learn better .
2. Less Exploration rate to explore but less randomness.
3. Average discount factor to give the average weightage to future rewards to be get.



1. Exploration will take more number of episodes to converge or learn the maze.
2. More Exploration rate , more episodes taken to converge
3. Less Exploration rate, less episodes taken to converge.
4. State Value which are near to goal state are lesser than the state value far from the goal state.Since, the State Value means the expected sum of rewards it will get from that state.So, the more near to goal state will get the less value
5. And the states which are the far from the goal state will get the more value as their expected sum of rewards will be more.

Q-Table Value after training the maze by the agent:

```
[[-0.0066210910079018154, -0.00661636714147668, -0.006498203574970651,
-0.0066210910079018154], [-0.006626727550515397, -0.0066393896007551495,
-0.006639335155576441, -0.006639335155576441], [-0.00664092884138266,
-0.0066393896007551495, -0.006639335155576441, -0.006639335155576441],
[-0.0066413197264958675, -0.00663937190607207, -0.006639335155576441,
-0.006639335155576441], [-0.006641712915988952, -0.006639335155576441,
-0.006639328349929104, -0.006639335155576441], [-0.006555146284315288,
-0.006554474380034223, -0.0060856788454847445, -0.006557433907399097],
[-0.006463646713589307, -0.006455346527025306, -0.006424239280744891,
-0.006455346527025306], [-0.004809993493270653, 0.005598090384796984,
-0.004817460445797728, -0.004809993493270653], [-0.0038021733569580237,
-0.0035706394214400005, 0.03819591799098862, -0.0035706394214400005],
[-0.006639335155576441, -0.006639335155576441, -0.006639335155576441,
-0.006639339218511301], [-0.006426530144346939, -0.006426530144346939,
-0.005119647461836924, -0.00643399442919702], [-0.006211631610655955,
-0.005931982306196383, -0.006211631610655955, -0.006181419332929828],
[-0.005784182321998611, -0.00568684097399925, -0.005683322536790367,
-0.003691425194156813], [-0.0021235200000000003, 0.13468166600079864,
-0.0021235200000000003, -0.002335479808], [-0.0008640000000000001,
-0.0015040000000000001, 0.38512103969798894, -0.0008],
[-0.006149581375745403, -0.002660419870529771, -0.006149581375745403,
-0.006182925281985885], [-0.005625354292515464, -0.00568684097399925,
0.0035197491391561137, -0.00568684097399925], [-0.00568684097399925,
-0.00568684097399925, -0.00568684097399925, -0.005689303880045468], [-0.0008,
0.3949734467793621, -0.0008, -0.0008], [-0.0008, -0.0008, 0.9974266761884933, 0],
[-0.00455681078780756, -0.004787123566942344, -0.00455681078780756,
-0.00455681078780756], [-0.004664184970207559, 0.018971053911490364,
-0.004809993493270653, -0.00482089025466779], [-0.0032744167014400003,
0.05756223509962209, -0.0035706394214400005, -0.0035706394214400005],
[-0.0021730816, -0.0021235200000000003, -0.0021235200000000003,
0.15398072067907403], [-0.006461578580757831, -0.006483554034363282,
-0.006442584032537864, -0.006468744093154887]]
```

Lesser value is toward the action at intermediate state which is nearer to it. And the states which are far away from the goal will have more state value as shown in Q-table.

4)

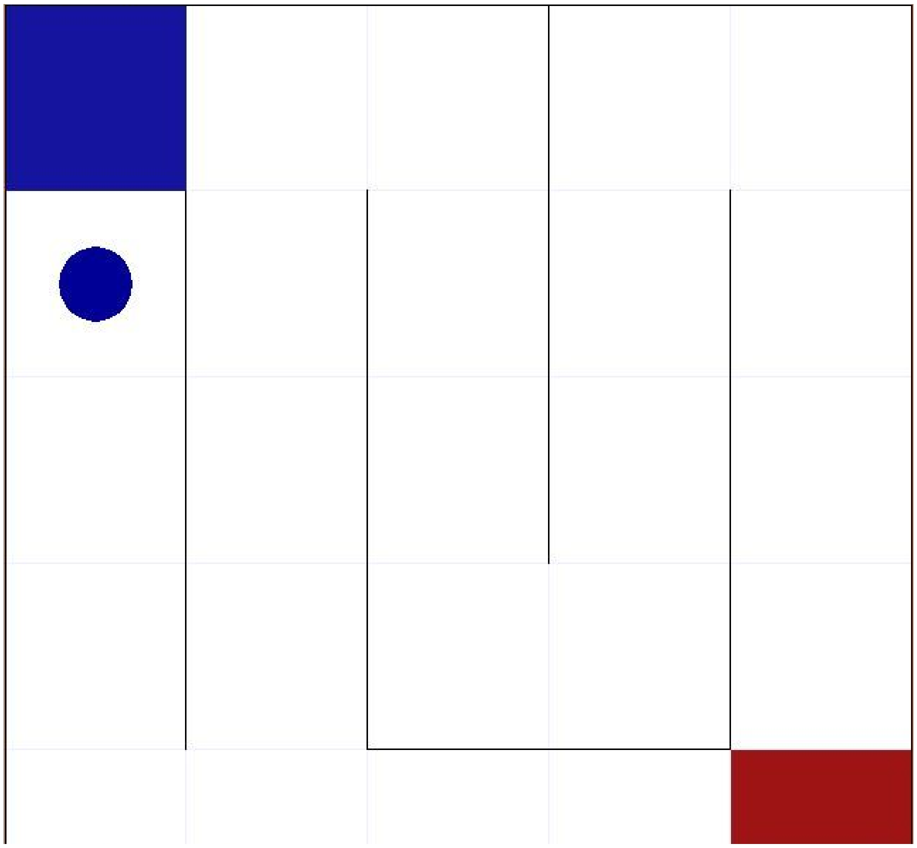
When created the new maze , then for agent it is new environment.

Agent get trained on the best hyperparameters value i.e, less learning rate , less exploration rate and average discount rate.

Since model get training from these hyperparameter and static environment .So, when the new maze will form, the agent will not trace the path from initial to final.

Performance:

1. In new maze, agent will get stuck at some cell position.
2. Agent will not trace the path from initial to final using the trained Q-table.



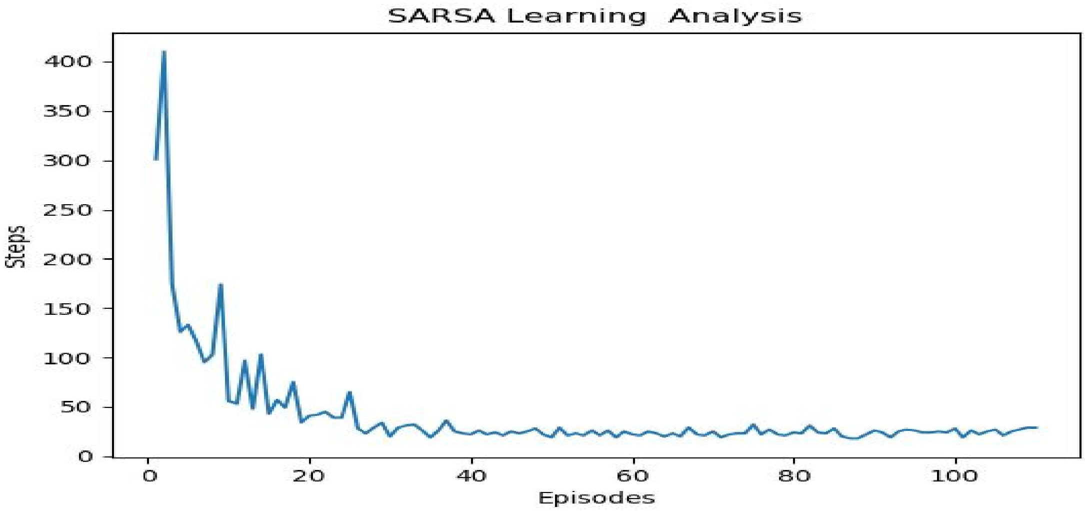
Solution for Problem:

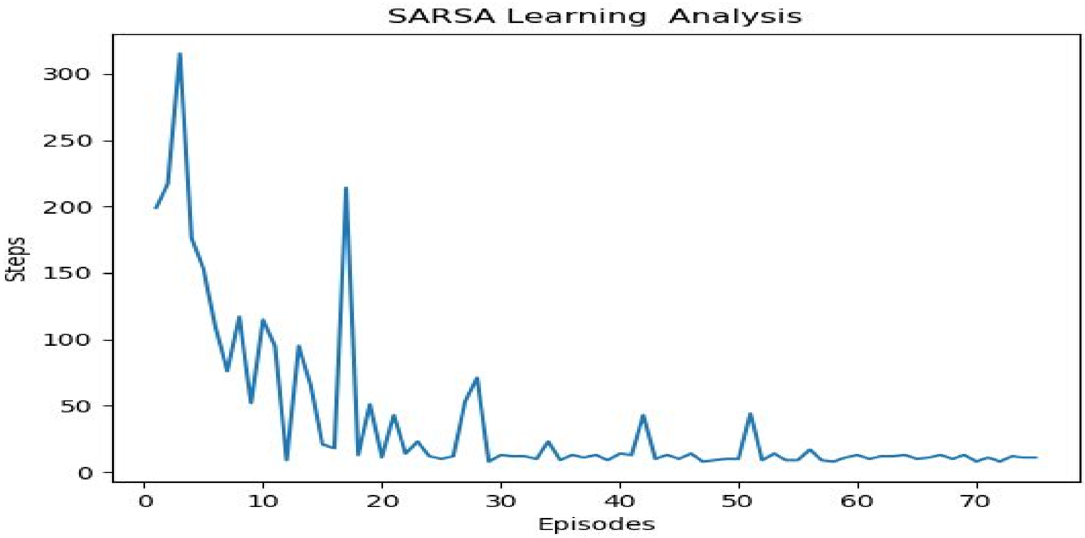
1. Use the exploration here again whenever it got stuck at the cell in maze.
2. Train again the agent onto new maze to make the agent trace the path from initial to goal state by agent.

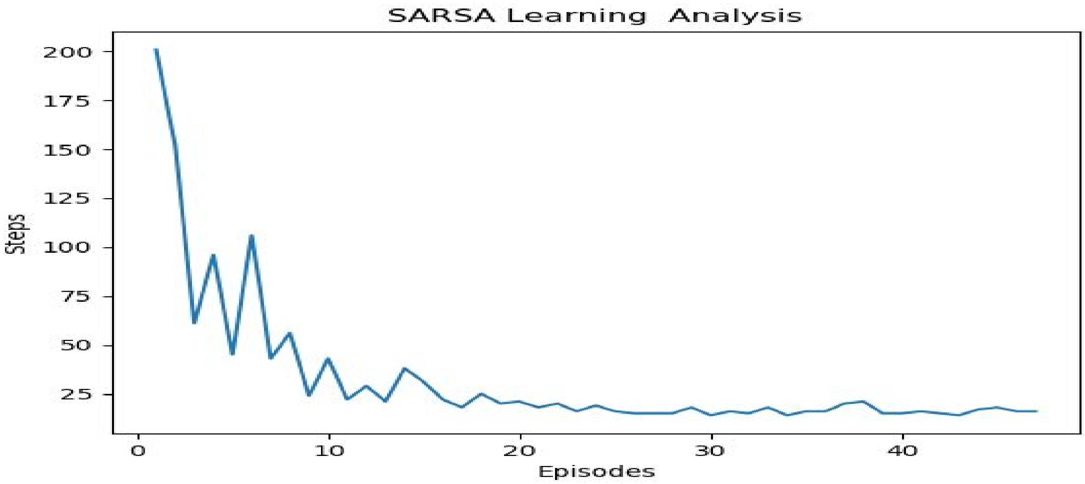
Bonus:

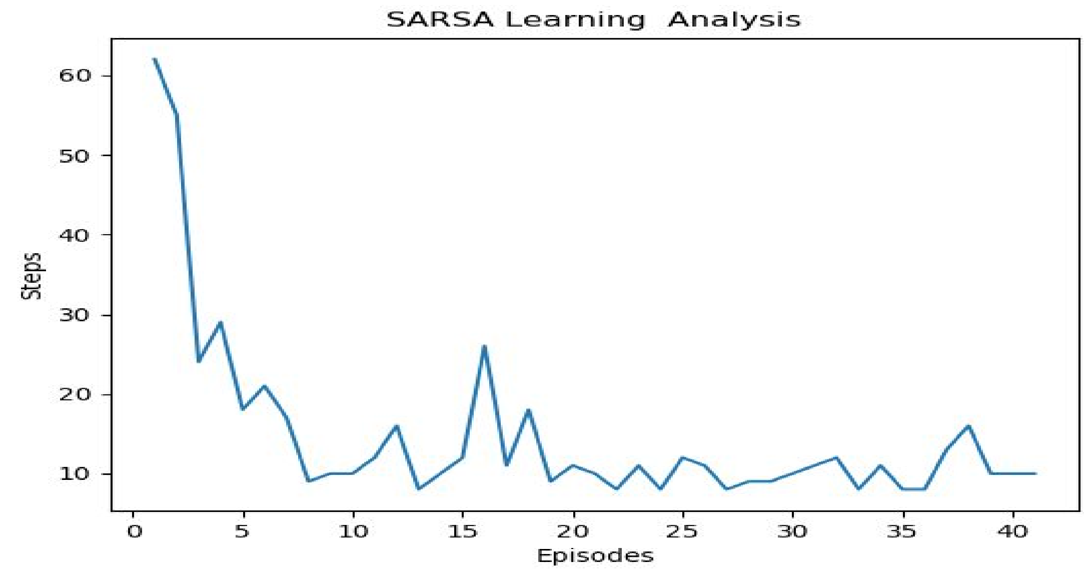
Q(st,at)=Q(st,at)+learning_factor*(reward_t+discount_factor*Q(st+1,at+1)-Q(st,at))

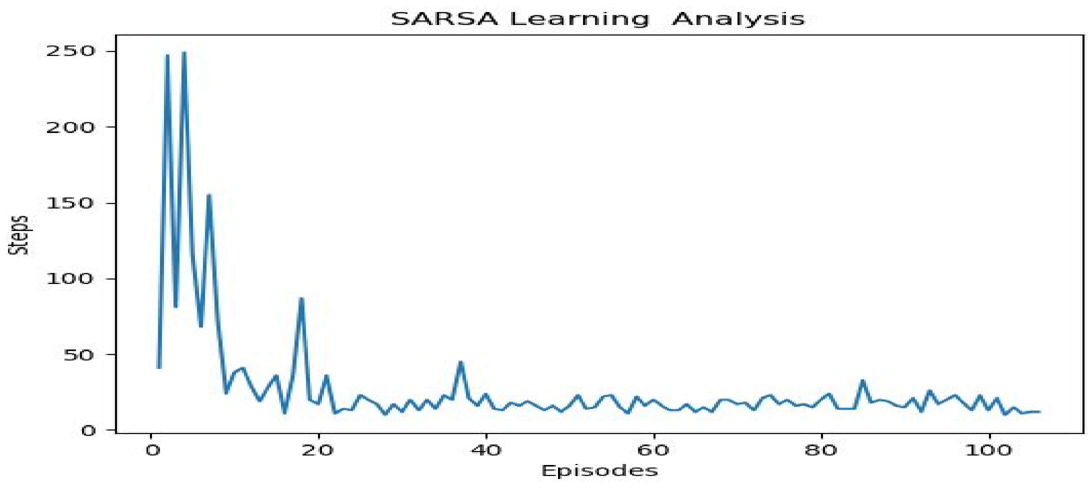
S.No.	Learning rate	Discount Factor	Exploration rate	Episodes	Time taken
1	0.4	0.3	0.4	58	43
2	0.4	0.3	0.2	40	28
3	0.4	0.5	0.2	105	74
4	0.2	0.2	0.2	105	74
5	0.6	0.4	0.1	70	49
6	0.8	0.5	0.2	42	24
7	0.2	0.4	0.6	36	25
8	0.2	0.8	0.4	155	65
9	0.2	0.8	0.8	101	52
10	0.2	0.4	0.01	53	49

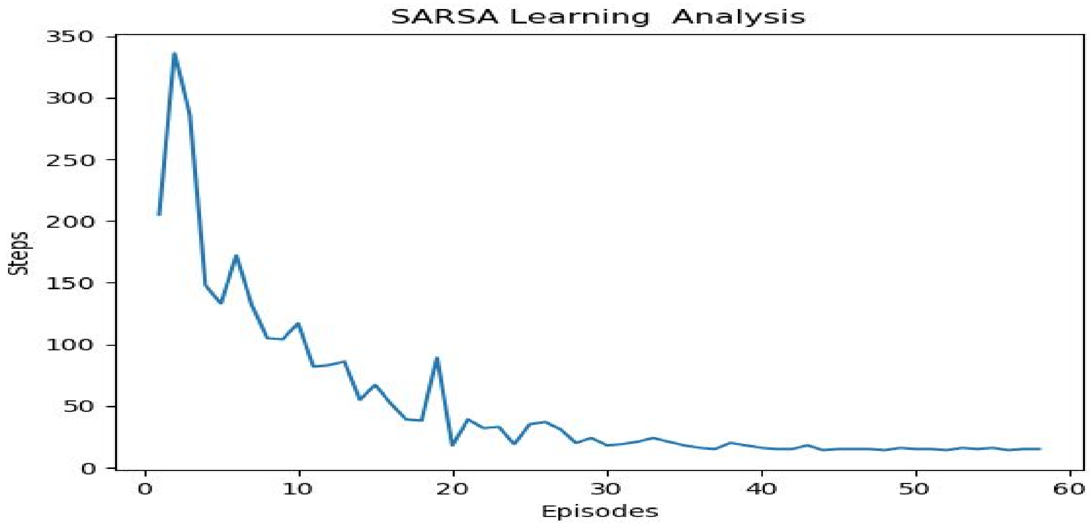

SARSA Learning Analysis
Steps
Episodes
0
50
100
150
200
250
300
350
400
0
20
40
60
80
100

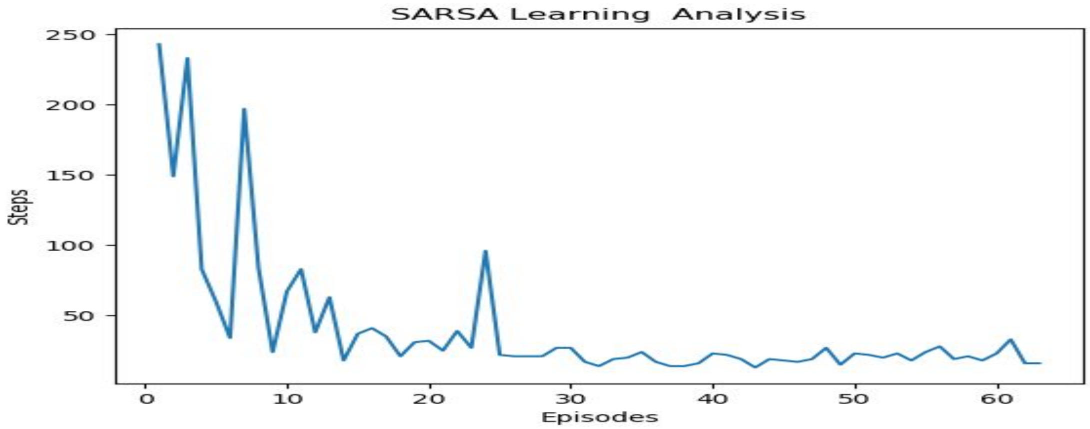

SARSA Learning Analysis
Steps
Episodes
0
50
100
150
200
250
300
0
10
20
30
40
50
60
70

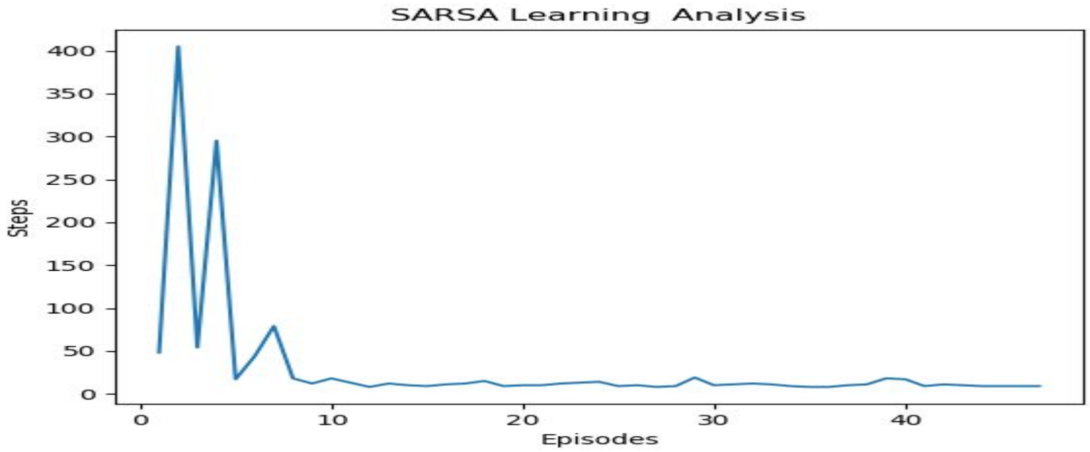

SARSA Learning Analysis
Steps
Episodes
25
50
75
100
125
150
175
200
0
10
20
30
40


SARSA Learning Analysis
Steps
60
50
40
30
20
10
0
5
10
15
20
25
30
35
40
Episodes


SARSA Learning Analysis
Steps
250
200
150
100
50
0
0
20
40
60
80
100
Episodes


SARSA Learning Analysis
Steps
350
300
250
200
150
100
50
0
0
10
20
30
40
50
60
Episodes


SARSA Learning Analysis
Steps
250
200
150
100
50
0
10
20
30
40
50
60
Episodes


SARSA Learning Analysis
Steps
400
350
300
250
200
150
100
50
0
0
10
20
30
40
Episodes

Conclusion:

- Q-learning directly learns the optimal policy, whereas SARSA learns a near-optimal policy while exploring
- Q-learning takes more time to converge than SARSA.
- From Observation, when discount factor is large then, the Q-learning algorithm will work more faster i.e, converges faster than SARSA.
- Number of Episodes taken to converge in SARSA will be small in comparison to Q-learning.And when the discount factor and exploration rate increases then, SARSA takes more number of episodes than Q-learning.