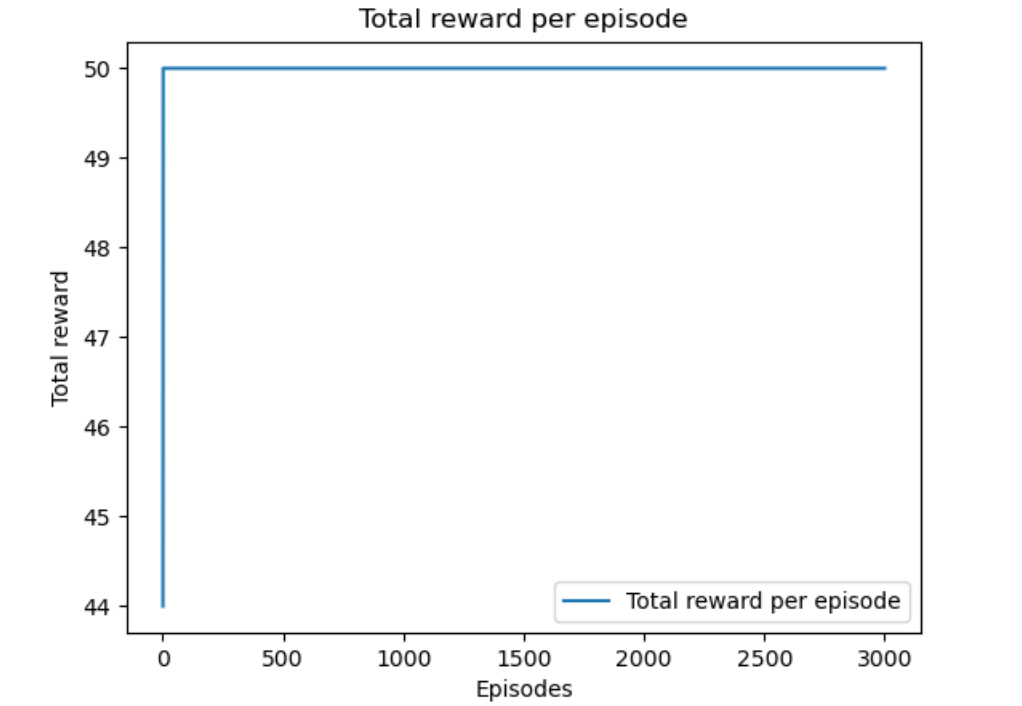
ASSIGNMENT 5

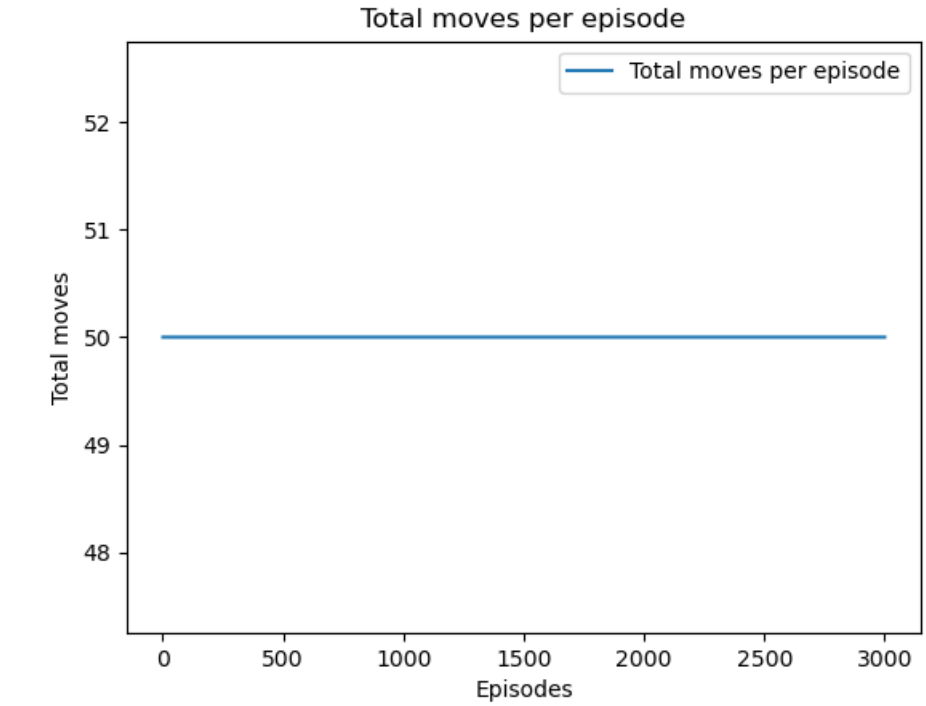
Christos Eleftheriou

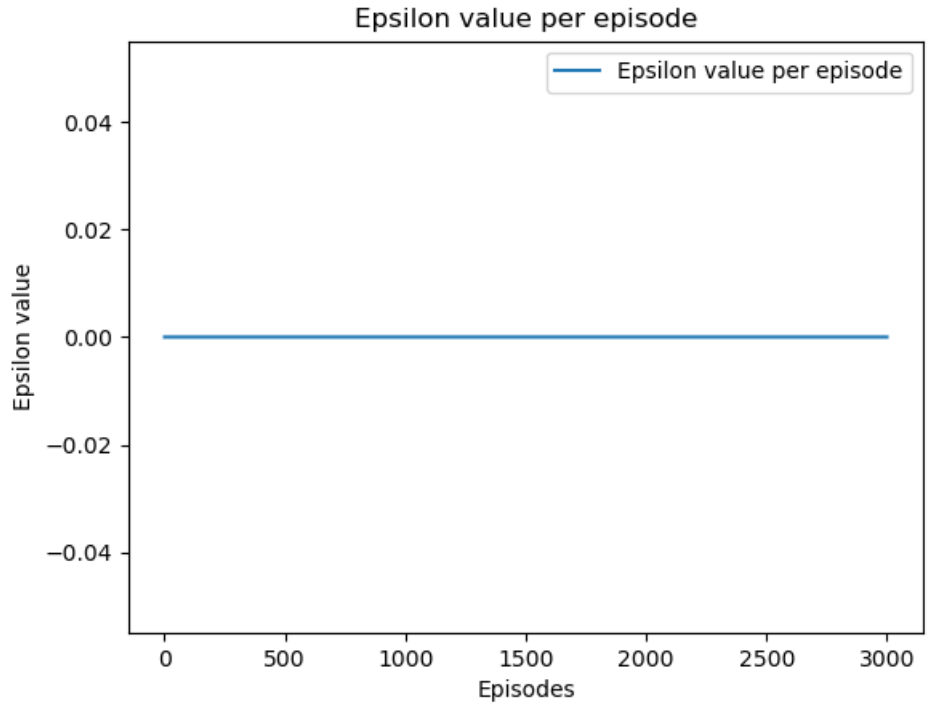
1009537

PART A

6.1




Total moves per episode
Total moves per episode
52
51
50
49
48
Total moves
0
500
1000
1500
2000
2500
3000
Episodes


Epsilon value per episode
Epsilon value per episode
0.04
0.02
0.00
−0.02
−0.04
Epsilon value
0
500
1000
1500
2000
2500
3000
Episodes

```
Final episode map
---------------------------------------------------
['X', 'X', 'X', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'O', 'X', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
---------------------------------------------------
```

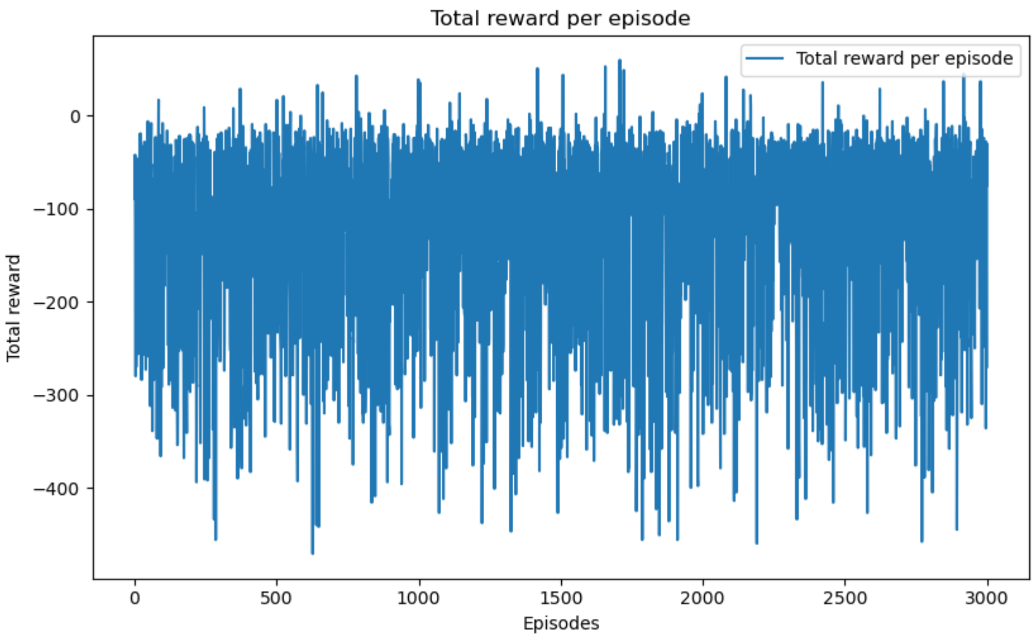
6.2

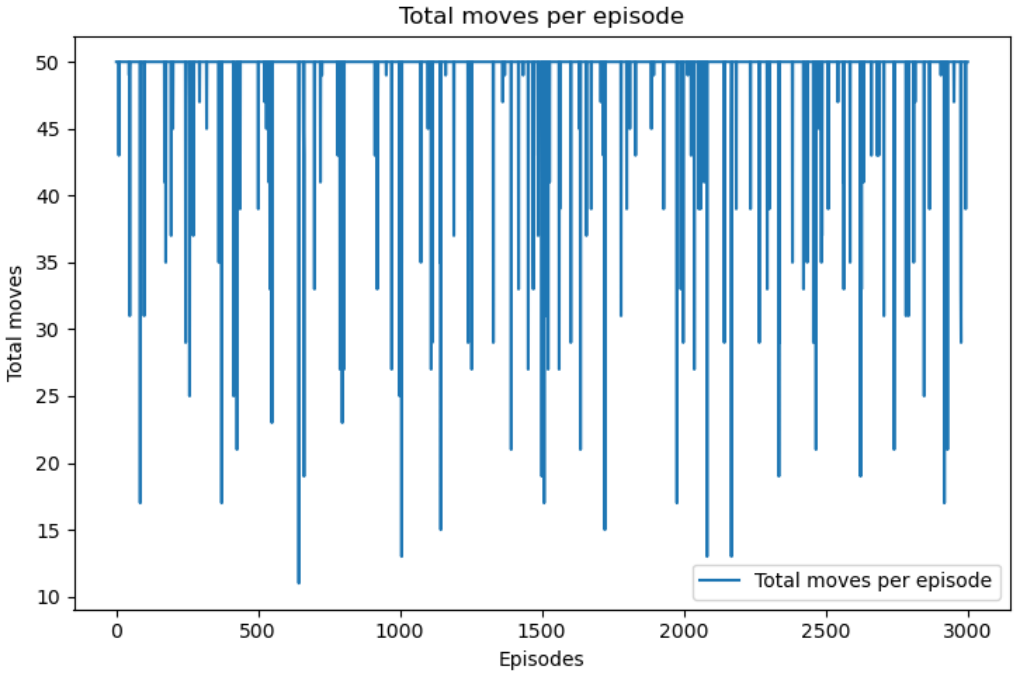
In the first plot we can see that the total reward for all episodes after is 50. This happens because as we can see from the map the agent reaches a point where it goes back and forth resulting in the reward getting +1 every time for 50 moves.

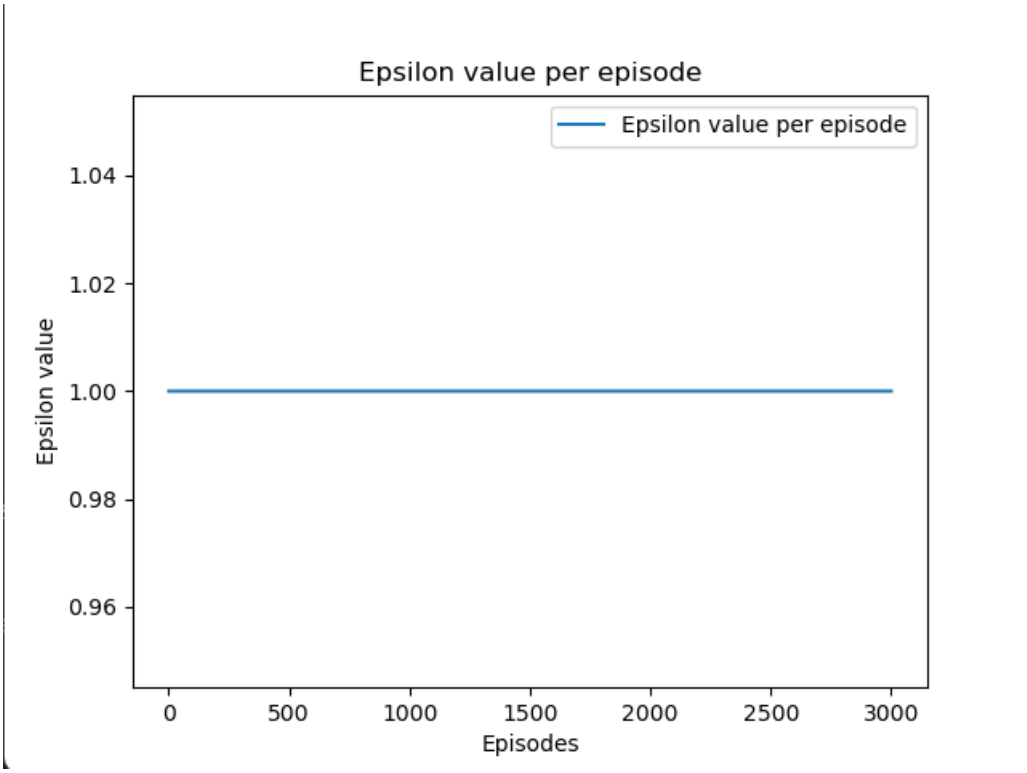
From the following 2 plots we can see that the total moves per episode are always 50 and the epsilon is 0 for all the episodes.

Because of the fact that epsilon is always 0, the agent does not use exploration in order to discover the optimal path to its destination. This results to the agent going back and forth between two places and not exploring further, and always doing 50 moves without finding the path.

7.1




Total moves per episode
Total moves
Episodes
Total moves per episode
50
45
40
35
30
25
20
15
10
0
500
1000
1500
2000
2500
3000


Epsilon value per episode
Epsilon value
Episodes
Epsilon value per episode
1.04
1.02
1.00
0.98
0.96
0
500
1000
1500
2000
2500
3000

```
Final episode map
--------------------------------------------------
['X', 'X', 'X', 'X', 'X', 'O', 'O', 'O', 'O', 'O']
['X', 'X', 'X', 'X', 'X', 'X', 'O', 'O', 'O', 'O']
['O', 'O', 'O', 'X', 'X', 'X', 'O', 'O', 'O', 'O']
['O', 'O', 'X', 'X', 'X', 'X', 'O', 'O', 'O', 'O']
['O', 'O', 'X', 'X', 'X', 'X', 'O', 'O', 'O', 'O']
['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
--------------------------------------------------
```

7.2

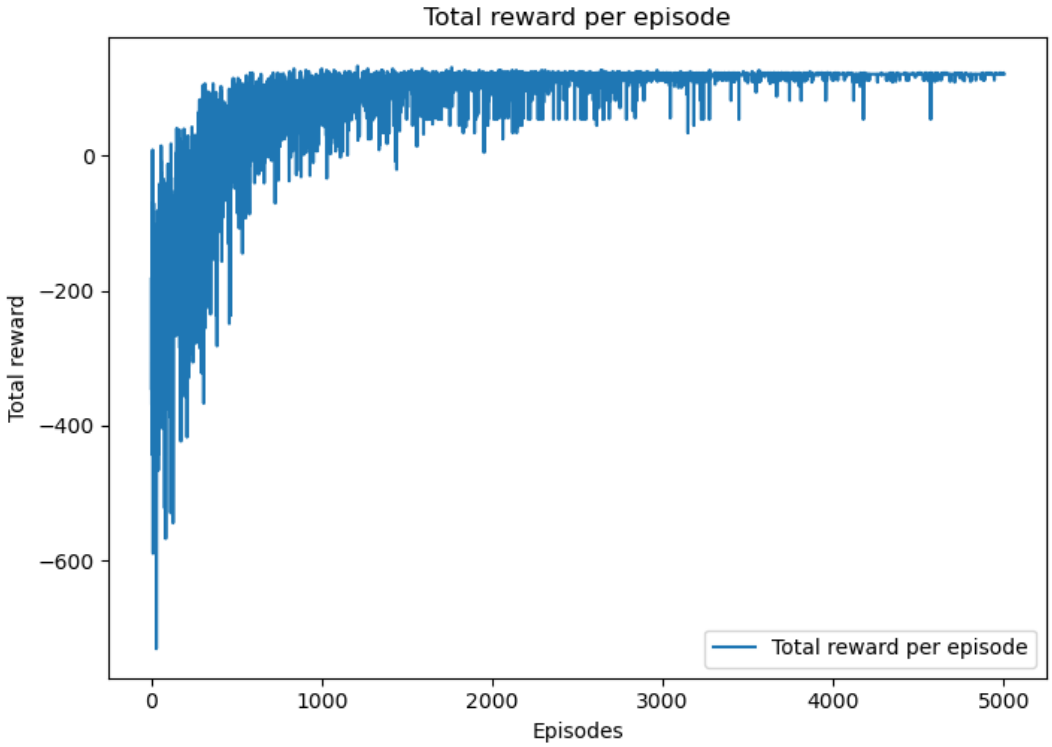
From the first plot we can see that the total reward fluctuates throughout all of the episodes. The same happens with the total number of moves, as we can see from the second plot.

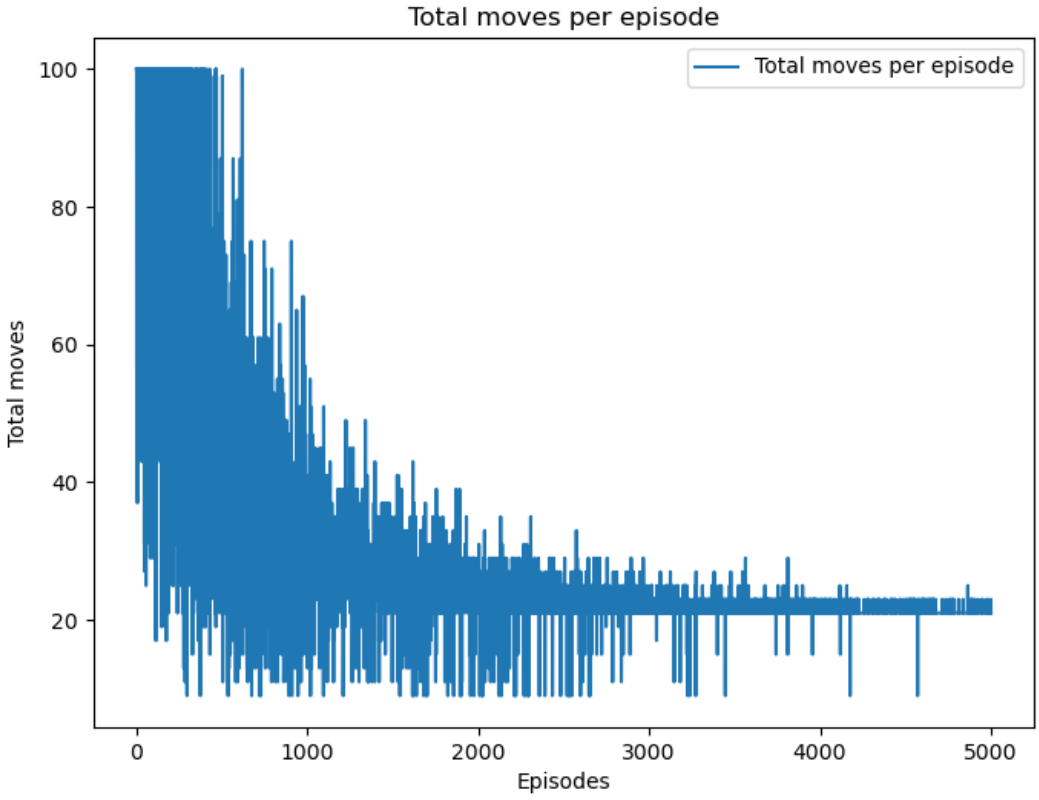
This happens because epsilon is always one, which can be seen at the third plot, meaning that the agent is always exploring randomly and never choosing the best choice from the q table. The epsilon decay rate is 0, so epsilon is one for all the episodes, which means that this random exploration happens for every episode. This results to the agent always exploring random new paths and not finding the optimal solution (this can happen on accident though).

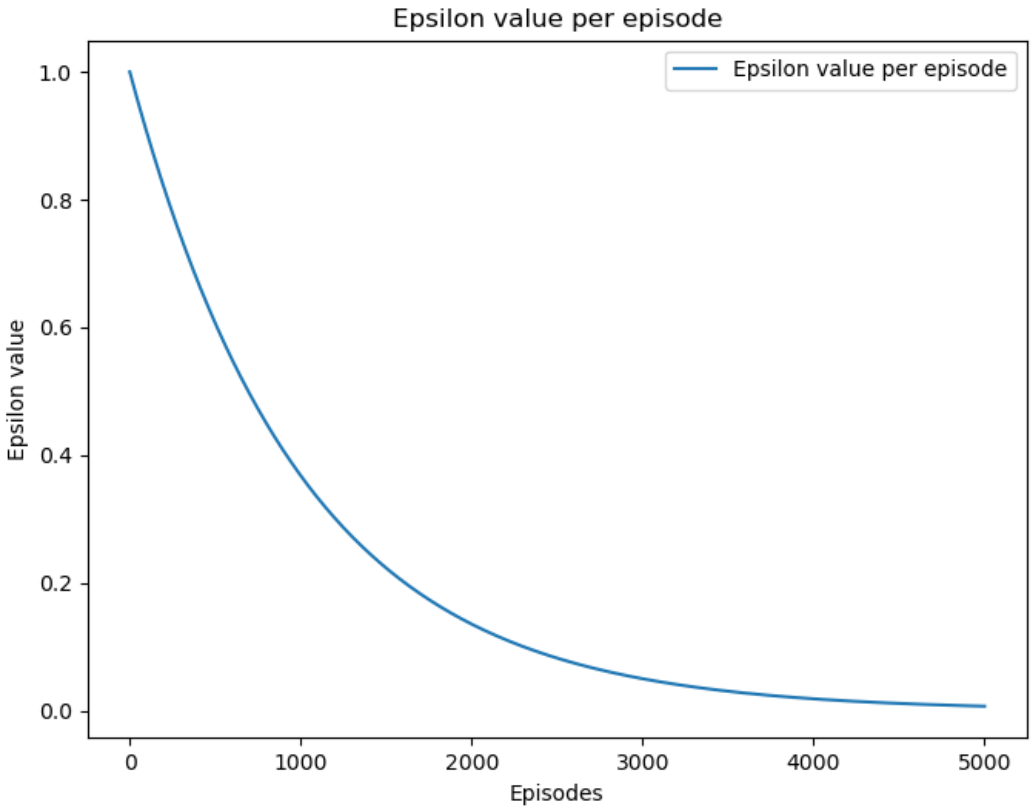
8.1

```python
#Best hyperparameters
run_q_learning(learning_rate = 0.9, gamma = 0.9, epsilon_decay_rate = 0.001, episodes = 5000, max_steps = 100, epsilon=1)
```

Despite the element of randomness in the agents decisions, using these hyperparameters I was able to produce the optimal path almost every time.


Total reward per episode
Total reward
0
−200
−400
−600
Total reward per episode
0
1000
2000
3000
4000
5000
Episodes


Total moves per episode
Total moves
100
80
60
40
20
Total moves per episode
0
1000
2000
3000
4000
5000
Episodes




```
Final episode map
--------------------------------------------------
['X', 'X', 'X', 'O', 'O', 'O', 'O', 'O', 'X', 'X']
['O', 'O', 'X', 'O', 'O', 'O', 'O', 'O', 'X', 'O']
['O', 'O', 'X', 'O', 'O', 'O', 'O', 'O', 'X', 'O']
['O', 'O', 'X', 'X', 'X', 'O', 'O', 'O', 'X', 'O']
['O', 'O', 'O', 'O', 'X', 'O', 'O', 'O', 'X', 'O']
['O', 'O', 'O', 'O', 'X', 'O', 'O', 'O', 'X', 'O']
['O', 'O', 'O', 'O', 'X', 'X', 'X', 'X', 'X', 'O']
['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
--------------------------------------------------
```

In the first plot we can see that the total reward increases as the episodes go by, and on the second plot we can see that the total number of moves decreases as the episodes go by. These results show that the performance of the agent improving throughout the episodes, as it learns to get the maximum reward with the fewer total moves (meaning finding the optimal path to the goal). This is an indication that the agent's learning process over time is correct, and on the final map we can see that the agent finds the optimal path to its goal. With the number of episodes set to 5000, in the initial episodes the agent uses exploration as well as the q table in order to make its decisions, but by the final episodes, the epsilon has decreased at a very low number meaning that the agent will

choose the optimal choice from the q table instead of randomness most of the time, leading to the agent always finding the optimal path.

On the third plot we can see that the value of epsilon is 1 initially and it decreases over time, as we have set the epsilon decay rate at 0.001.

8.2

```
Episode 5000/5000, Epsilon: 0.006721111959865607, Total Reward: 120, Moves: 21
```

8.3

Yes there are slower paths that the agent could have followed where it could have accumulated a better reward. This path could be the path that leads to the cell (7,2), and then it returns back and goes to the goal cell (0,9).

The agent did not follow this path because we have a high learning rate meaning that the agent will converge quicker using the initial path it found. We also have a decreasing value of epsilon which means that over time the agent will reduce exploring and it will prefer to exploit known paths. The high gamma value also contributes to this, as it means that the agent prefers more immediate rewards than future rewards, so it will follow the path that will give the reward faster than the one that will give the reward in the future.

PART B

The basic difference between the q-learning algorithm and the SARSA algorithm is in the way that the q-values are updated for each one of them.

1.

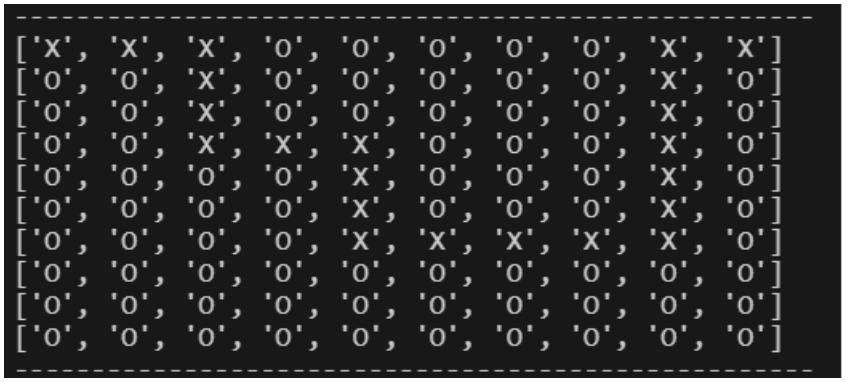
For a static ε value of 0.1 a possible path SARSA could have generated is this:

```
--------------------------------------------------
['X', 'X', 'X', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'X', 'X', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'X', 'X', 'X', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'X', 'X', 'X', 'X', 'O', 'O', 'O', 'O', 'O']
['O', 'O', 'X', 'X', 'X', 'O', 'O', 'O', 'O', 'O']
['O', 'O', 'X', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'O', 'X', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'O', 'X', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
--------------------------------------------------
```

As we can see the agent did not find the path to the goal but instead followed a wrong one. This is because the epsilon has a small and static value of 0.1 which means there is only ten percent chance of random explorations, and this means that the agent does not explore effectively. And because the epsilon value does not decrease, the agent does not end up making the best decisions always, which leads to a path that does not lead to the goal.

2.

For a decreasing ε value of 0.1 a possible path SARSA could have generated is this:



As we can see the agent will find the path to the goal. While the ε value might be a bit small not allowing a lot of exploration, the fact that the epsilon decreases makes the agent learn and find the best path, because it makes the optimal decisions over time by learning.