City, University of London

Reinforcement Learning Coursework

Autonomous Maze Solving Agent

Author:
Chadi-El-Hajj
Muaaz BinSarfaraz

Supervisor:
Dr. Eduardo Alonso

March 26, 2017

Contents

1 Introduction **3**

2 Domain representation and task **3**
2.1 Transition functions 4
2.2 Graphical representation of the domain 5
2.3 Reward Function 5
2.4 R-Matrix 5

3 Q-learning **5**
3.1 Q-learning steps 7
3.1.1 Q-learning variables 9

4 Policy π **9**

5 Parameters configuration used for Q-learning **10**

6 Case 1 **11**
6.1 Q-learning solved by hand 11
6.2 Performance Analysis 13

7 Case 2 **15**

8 Case 3 **16**

9 Case 4 **17**

10 Case 5 **18**
10.1 Case 5A slight changes in Reward and State 18
10.2 Case 5B extreme changes in Reward and State 21

11 Conclusion **23**

List of Figures

1 Representation of the domain by a 7x7 grid maze. Blue square represents the starting state and the green square represents the goal state. Black squares are walls/obstacles. 3
2 From left to right. The first figure represents the shortest path. The middle figure represents the medium path and last figure on the right is the longest path. 4
3 The original maze showing the loop where the agent gets stuck. Loop: 26, 25,24,23,30,31,32,33. 4
4 Graphical representation of the transitions states. 7
5 Original immediate reward matrix. 8
6 Initial Q-matrix. 12
7 Q-matirx after one episode. 13
8 Updated Q-matrix after 3861 episodes. 14
9 Converged Final Q-matrix. 15
10 learning performance- case 1. Epsilon 0.9, Gamma 0.8 and alpha 0.8. 15
11 Left: Q-learning performance- Case 2. Gamma 0.10. Right: Gamma 0.50 16
12 Left: Q-learning performance- Case 3. Alpha 0.20. Right: Alpha 0.50 16
13 Q-learning performance- Case 4. Epsilon 1 (random policy) 17
14 Penalised maze case 5. Blue is the initial start state, pink represents the obstacle and green is the goal state 18
15 New representation of the domain with one penalised state 19
16 Q-learning performance- Case 5A 20
17 Case 5B agent is allowed to enter pale green wall with a penalty of -5, black walls are not allowed and green wall has higher penalty of -25. Reward of reaching final goal state-25 is 1000 points 21
18 Graph with updated rewards for Case 5B 22
19 Graph with updated rewards for Case 5B 23

List of Tables

1 Transition State Table. 6

1 Introduction

Reinforcement learning is a machine learning technique which makes it a branch of Artificial Intelligence. It allows the software agents to move autonomously and make their own decisions to determine the optimal behaviour in an unknown and unpredictable environment. Unlike most machine learning techniques where the machine is trained by a set of examples, the software agent must learn on its own to take the ideal actions that maximises the cumulative reward signals without any knowledge about the environment. Rewards in turn determine the performance of the agent which helps it learn the unknown environment [1][2].

Many reinforcement learning algorithms were developed to support the idea mentioned above e.g. Monte Carlo, Eligibility trace and Q-learning. In this report, we will use Q-learning algorithm to solve reinforcement learning problem. Q-learning will allow the agent to learn by experiment. The agent starts from an initial state and traverse from state to another until it reaches the goal state. This process of moving from initial state to the goal state is called an episode and it is repeated many times until the agent learns all the paths. In this study, we will show how the algorithm works, experiment with different parameter values e.g. learning rate and discount rate. Also we will experiment using different policy, and reward matrix. Finally we will compare and analyse the results of the experiments.

2 Domain representation and task

The chosen domain in this report is an agent/robot that starts from an initial state and has to park itself in its designated spot consuming the least amount of energy, hence it has to find the shortest path to get to its destination. The model is represented by a 7x7 grid (Figure 1) which results in 49 different states. Each square in the grid denotes a state. Each state has different actions that the agent can take to move to other state. The agent starts from an initial state (coloured blue)and has to traverse the maze to find its goal state(coloured green). It can move from the blue state though any white state, however, it cannot jump through walls denoted as a black squares in figure 1, nor move outside the grid.

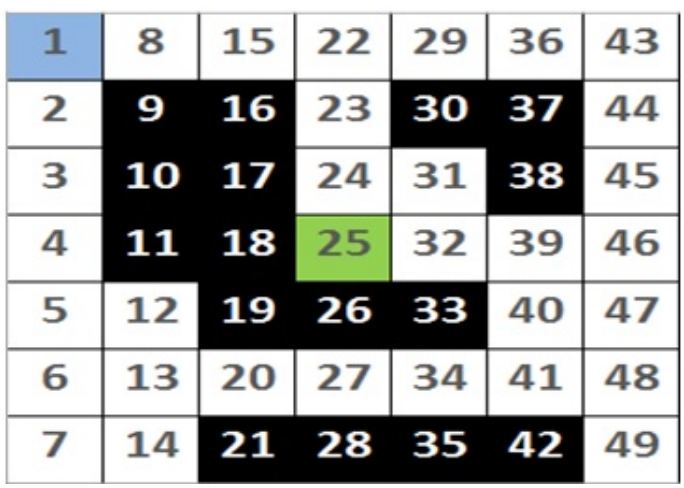

Figure 1: Representation of the domain by a 7x7 grid maze. Blue square represents the starting state and the green square represents the goal state. Black squares are walls/obstacles.

Figure 2 below shows three examples routes that the agent might take. The agent starts at the blue state and can traverse the white state in which it will be awarded zero since it is not the goal

state. If the agent cannot traverse to the black state/wall, it will receive a negative infinity. And when agent reaches the goal state,it will be awarded 500.

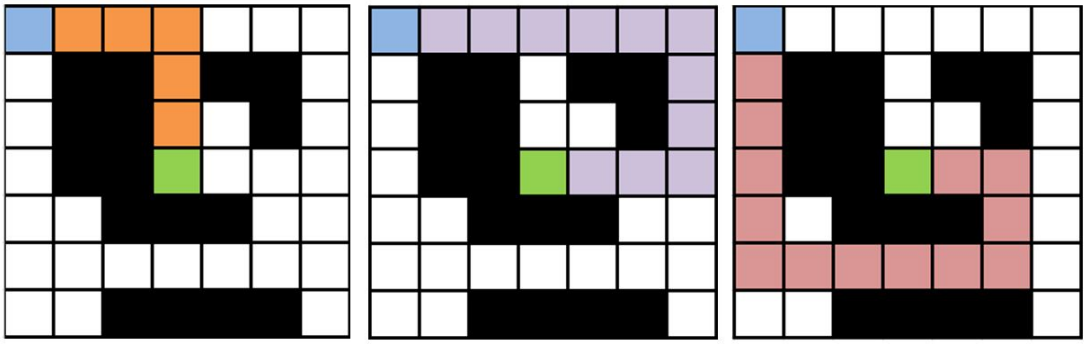

Figure 2: From left to right. The first figure represents the shortest path. The middle figure represents the medium path and last figure on the right is the longest path.

Note: the maze presented in Figure 1 is an alternative (amended) of the original maze shown in Figure 3 below. This maze had to be changed since the agent gets stuck in a loop most of the time and code gets trapped in a certain episode for a long time. Therefore, we change it to the maze presented in Figure 1 to avoid the risk of entering an infinite loop and make sure that the code runs and the agent reach its destination every time.

1	8	15	22	29	36	43
2	9	16	23	30	37	44
3	10	17	24	31	38	45
4	11	18	25	32	39	46
5	12	19	26	33	40	47
6	13	20	27	34	41	48
7	14	21	28	35	42	49

Figure 3: The original maze showing the loop where the agent gets stuck. Loop: 26, 25,24,23,30,31,32,33.

2.1 Transition functions

At a given state S, the agent has a set of actions A that it can take to move to another state S'. Once the agent gets to S', it becomes the current state that has a different defined set of actions itself. The agent can move up, down, left and right inside the grid only. Which also means it can move from state S to S' and from S' back to S. However, it cannot move diagonally, nor jump through walls presented as black squares. The agent starts from the initial blue state and moves across the white

allowed states to get to the goal green state. The possible transition functions of all states are shown in Table 1.

2.2 Graphical representation of the domain

Figure 4 below is a graphical representation of the problem. The graph shows the start and finish state along with the transition states. A link is drawn between the allowed states to denote the path that the agent can take. The red states represent the wall where the agent cannot pass and there is no links between these states to any other states. The numbers shown between the links indicate the reward that the agent will receive when traversing between the states. The rewards are set to zero when the agent moves between the allowed states and 500 only when the agent get to the green goal state – number 25. The agent will only receive 500 if it goes from state 24 to 25 or from 32 to 25 or loop on the 25th.

2.3 Reward Function

The reward is the numerical value that the agent will receive when traversing the states. It sets the desirability of the state. As stated earlier in the introduction, the agent's goal is to maximise the cumulative rewards. The agent will not always move to states with highest reward but it will prefer the path in which it will receive highest cumulative rewards instead [1]. Figure 4 above shows the rewards that the agent will receive when moving between states. The agent is allowed to traverse from the initial state to the goal state through the white states receiving zero reward. It will be awarded 500 when it gets to the final state. Moreover, the agent is not allowed to move to the red states where the rewards are set to minus infinity(-inf) to deny the agent from moving through the wall.

2.4 R-Matrix

The R matrix in Figure 5 shows the immediate reward at each state. It shows zero rewards where the agent can pass and minus infinity (-inf) where there is no link and therefore the agent is not allowed to pass. As stated before, there are three possible moves that the agent can take to get the 500 rewards, the three possible moves are shown in Figure 5

3 Q-learning

The reinforcement learning algorithms chosen to solve this problem is the q-learning. This algorithm will store the reward values, in a Q-matirx, after each time the agent moves to another state. We initialised the values in the Q-matrix to zero since the agent does not have any knowledge about the environment at the beginning. The agent will start from the initial state, find its way to the final state, and updates the Q-matrix as it progresses. We will demonstrate how the Q-matrix gets updated by showing the first episode written by hand, later in this report.

In order to solve this Q-learning problem, we used the Q-learning algorithm implemented by KardiTeknomo in matlab [3]. The code was amended to fit our specific task and support our experiment.

State S	Next possible states S'
1	{2,8}
1	{2,8}
2	{1,3}
3	{2,4}
4	{3,5}
5	{4,6,12}
6	{5,7,13}
7	{6,14}
8	{1,15}
12	{5,13}
13	{6,12,14,20}
14	{7,13}
15	{8,22}
20	{13,27}
22	{15,23,29}
23	{22,24}
24	{25,31}
25	{24,32}
27	{20,34}
29	{22,36}
31	{24,32}
34	{27,41}
36	{29,43}
39	{32,46}
40	{39,41,47}
41	{40,48}
43	{36,44}
44	{43,45}
45	{44,46}
46	{39,45,47}
47	{40,46,48}
48	{41,47,49}
49	{48}

Table 1: Transition State Table.

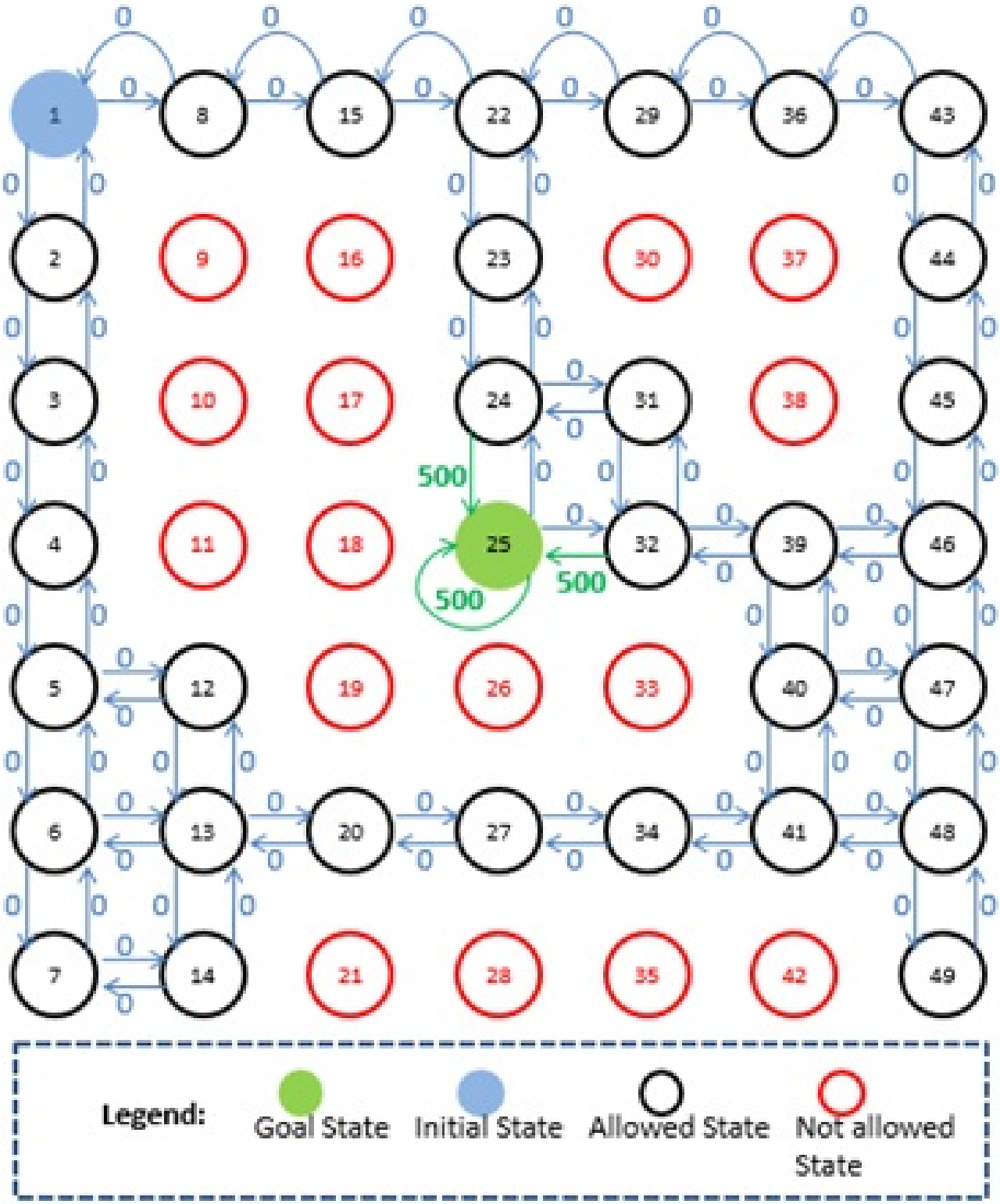

Figure 4: Graphical representation of the transitions states.

3.1 Q-learning steps

Initialise a 49 x 49 null Q-matrix,
Initialise a 49 x 49 R-Matrix using graph diagram given in Figure 4

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
1	-	0	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	0	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	-	0	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4	-	-	0	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	-	0	-	0	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	0	-	0	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	0	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	0	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	0	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	0	-	-	-	-	-	0	-	0	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	0	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	0	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	500	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
25	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	500	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
26	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
27	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
28	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
29	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-
30	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
31	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
32	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	500	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-
33	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
34	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-
35	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
36	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-
37	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
38	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
39	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	0	-	-	-	-	-	0	-	-	-
40	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	0	-	-	-	-	-	0	-	-
41	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	0	-	-	-	-	-	-	-	0	-
42	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
43	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	0	-	-	-	-	-
44	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	0	-	-	-	-
45	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	0	-	-	-
46	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	0	-	0	-	-
47	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	0	-	0	-
48	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	0	-	0
49	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-

Figure 5: Original immediate reward matrix.

Set γ Gamma , α Alpha and ϵ Epsilon constants
Start from an initial state position (State 1)

Loop While goal state not reached:

1. Choose an action from possible actions using relevant policy
2. Pick the reward value $r(s_t, a_t)$ value from R-matrix
3. Pick the $Q(s_t, a_t)_{old}$ value from Q-matrix
4. Assuming action is performed, list all possible next state $Q(s_{t+1}, a')$ values

5. Select the maximum $Q(s_{t+1}, a)$ value from the list
6. Using the values collected in previous steps, calculate $\rho = [r(s,a) + \gamma maxQ(s_{t+1}, a')] - Q(s,a)_{old}$
7. Calculate: $NewQ(s_t, a_t) = Q(s_t, a_t)old + \alpha\rho$
8. Update Q-matrix overwriting $NewQ(s_t, a_t) on Q(s_t, a_t)_{old}$
9. The new state will become the current state

End Loop

3.1.1 Q-learning variables

-$Q(s_t, a_t)_{old}$ stands for the value of the old state and action

-a_t stands for action

-s_t stands for state

-$r(s_t, a_t)$ stands for the reward

-$maxQ(s_{t+1}, a')$ stands for the max rewards amongst the next possible states and their actions

-α is the learning rate. It takes a value between 0 and 1. This value indicates how fast the Q-matrix values get updated as the agent traverses between the states. In other words, α represents how fast the agent learns the new values and replace the old values. On one hand, the higher the α value is, the quicker it will learn and update the old ones but it will not necessary give accurate results. On the other hand, if the value is set to zero then it will not updated the old Q-matrix values

-γ is the discount factor. It takes a value between 0 and 1. This values indicates the importance of the future rewards. If the value is closer to 1 then the future/cumulative rewards matter more than the immediate reward. Consequently, if value is small and closer to zero then it favours the immediate rewards, thus the agent will always move to the state with the higher reward values[4].

4 Policy π

The policy chosen for this problem is the ϵ-greedy policy.ϵ takes a value between 0 and 1. A high ϵ value means there is a higher probability that the next action will be chosen at random as opposed to a small ϵ value where the agent will take the action which returns the highest rewards. In other words, if ϵ value is big then there is higher probability that the agent will explore to find the best/shortest paths, otherwise with a smaller ϵ value it will exploit by choosing the best known path[5].

E-greedy policy algorithm:

- Set ϵ value to 0.9
- Randomly select a value (RD) between 0 and 1 inclusive from a uniformed distribution.
- If RD $>= \epsilon$ then choose the next state with the highest rewards

- Else randomly select the next state
- End If
- If $\epsilon >= 0.5$ then
- $\epsilon = \epsilon * 0.99999$
- Else
- $\epsilon = \epsilon * 0.9999$
- End If

5 Parameters configuration used for Q-learning

Five different cases are studied in this report.

Case 1: The values chosen for γ (Gamma), α (Alpha) and ϵ (epsilon) are as follows:

- Gamma: 0.80
- Alpha: 0.80
- Epsilon: 0.90
- Policy is epsilon greedy
- Goal rewards: 500

Case 2: Changing discount factor γ (Gamma) value:

- Gamma: [0.10, 0.5]
- Alpha: 0.80
- Epsilon: 0.90
- Policy is ϵ-greedy
- Goal rewards: 500

Case 3: Changing learning rate, α (Alpha) value:

- Gamma: 0.8
- Alpha: [0.20,0.5]
- Epsilon: 0.90
- Policy is ϵ-greedy
- Goal rewards: 500

Case 4: Changing policy, random and greedy policy:

- Gamma: 0.8
- Alpha 0.8
- Epsilon: 0.90
- Policy: random policy , ϵ =1
- Goal rewards: 500

Case 5: Changing the maze and the reward:

- Gamma: 0.80
- Alpha: 0.80
- Epsilon: 0.90
- Policy is ϵ-greedy
- Goal rewards: 1000

6 Case 1

6.1 Q-learning solved by hand

Calculation for complete 1 episode shown below:
$\alpha = 0.8\gamma = 0.8\epsilon = 0.9$

1st Step
Current State 1, Action 1 : Right, next state 8
$Q(1,8)_{new} = Q(1,8) + \alpha[r(1,8) + \gamma max\{Q(8,15), Q(8,1)\} - Q(1,8)]$
$= 0 + \alpha[0 + \gamma max\{0,0\} - 0] = 0$

2nd Step
Current State 8, Action 2: Right, next state 15
$Q(8,15)_{new} = Q(8,15) + \alpha[r(8,15) + \gamma max\{Q(15,8), Q(15,22)\} - Q(8,15)]$
$= 0 + \alpha[0 + \gamma max\{0,0\} - 0] = 0$

3rd Step
Current State 15, Action 3: Right, next state 22
$Q(15,22)_{new} = Q(15,22) + \alpha[r(15,22) + \gamma max\{Q(22,23), Q(22,29), Q(22,15)\} - Q(15,22)]$
$= 0 + \alpha[0 + \gamma max\{0,0\} - 0] = 0$

4th Step
Current State 22, Action 4: Down, next state 23

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6: Initial Q-matrix.

$Q(22,23)_{new} = Q(22,23) + \alpha[r(22,23) + \gamma max\{Q(23,24), Q(23,22)\} - Q(22,23)]$
$= 0 + \alpha[0 + \gamma max\{0,0\} - 0] = 0$

5th Step

Current State 23, Action 5: Down, next state 24
$Q(23,24)_{new} = Q(23,24)) + \alpha[r(23,24) + \gamma max\{Q(24,25), Q(24,31), Q(24,23)\} - Q(23,24)]$
$= 0 + \alpha[0 + \gamma max\{0,0\} - 0] = 0$

6th Step

Current State 24, Action 6: Down, next state 25
$Q(24,25)_{new} = Q(24,25) + \alpha[r(24,25) + \gamma max\{Q(25,32), Q(25,24)\} - Q(24,25)]$
$= 0 + \alpha[500 + \gamma max\{0,0\} - 0] = \alpha[500] = 0.8x500 = 400$

Figure 7 shows the updated Q-matrix after performing the first episode on initial Q-matrix Figure 6. This matrix shows the rewards the agent got when it moved from state 24 to the goal state 25.

The value is 400 as shown in the hand written example above where the learning rate is set to 0.8.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	400	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 7: Q-matirx after one episode.

In the below matrix in Figure 8 we show the values of the updated Q-matrix after 3861 episodes. Similarly the matrix in Figure 9 shows the converged Q-matrix after 9493 episodes.

6.2 Performance Analysis

Figure 10, shows that initially it took the agent more than 180 steps to reach the goal state since it did not have any prior knowledge about the environment. Also, at the start, epsilon value was high, which means that the agent was exploring to learn the maze more than exploiting. This exploration at the beginning can be seen by the high fluctuation in the return per steps (red lines Figure 10). As the number of episodes increased, the agent is acquiring more knowledge about the environment, therefore the number of steps taken to reach the destination decrease too. It is evident that by the increased number of episodes the agent tend to exploit more, leading to less fluctuation and higher return per steps. Moreover, the less fluctuated red line proves that the agent had less to learn towards the end as it exploited the knowledge of all the paths to the goal state.The graph shows that the agent has converged and reached an optimal asymptotic level after around 4000 episodes and the

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
1	0	21	0	0	0	0	0	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	26	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	21	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	17	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	13	0	13	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	11	0	11	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	13	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	26	0	0	0	0	0	0	0	0	0	0	0	0	0	41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	11	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	13	0	0	0	0	0	13	0	13	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	11	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	33	0	0	0	0	0	0	0	0	0	0	0	0	0	51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	41	0	0	0	0	0	0	0	64	0	0	0	0	0	41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	51	0	80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	64	0	100	0	0	0	0	0	64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	51	0	0	0	0	0	0	0	0	0	0	0	0	0	33	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80	0	0	0	0	0	0	0	80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	64	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0	0	0	0	0	0	0	0	0	0	0	41	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	41	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80	0	0	0	0	0	0	0	51	0	0	0	0	0	51	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	64	0	41	0	0	0	0	0	41	0	0
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33	0	0	0	0	0	51	0	0	0	0	0	0	0	33	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33	0	0	0	0	0	0	0	33	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0	41	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33	0	51	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	64	0	0	0	0	0	41	0	41	0	0
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	51	0	0	0	0	0	51	0	33	0
48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	41	0	0	0	0	0	41	0	26
49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33	0

Figure 8: Updated Q-matrix after 3861 episodes.

number of steps stabilised at approximately 6 steps.
Note: exploitCount = 59031 exploreCount = 44925

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
1	0	21	0	0	0	0	0	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	26	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	21	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	17	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	13	0	13	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	11	0	11	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	13	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	26	0	0	0	0	0	0	0	0	0	0	0	0	0	41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	11	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	13	0	0	0	0	0	13	0	13	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	11	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	33	0	0	0	0	0	0	0	0	0	0	0	0	0	51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	41	0	0	0	0	0	0	0	64	0	0	0	0	0	41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	51	0	80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	64	0	100	0	0	0	0	0	64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	51	0	0	0	0	0	0	0	0	0	0	0	0	0	33	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80	0	0	0	0	0	0	0	80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	64	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0	0	0	0	0	0	0	0	0	0	41	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	41	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80	0	0	0	0	0	0	0	51	0	0	0	0	51	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	64	0	41	0	0	0	0	0	41	0	0
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33	0	0	0	0	0	51	0	0	0	0	0	0	0	33	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33	0	0	0	0	0	0	0	33	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0	41	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33	0	51	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	64	0	0	0	0	0	41	0	41	0	0
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	51	0	0	0	0	0	51	0	33	0
48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	41	0	0	0	0	0	41	0	26
49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33	0

Figure 9: Converged Final Q-matrix.

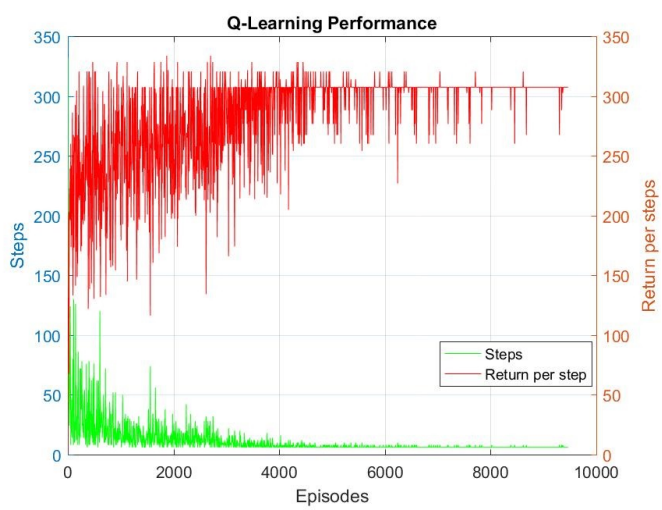

Figure 10: learning performance- case 1. Epsilon 0.9, Gamma 0.8 and alpha 0.8.

7 Case 2

Using a smaller discount factor (Gamma), Figure 11 shows that the smaller Gamma value yields to a smaller return per step. Since discount factor Gamma sets the importance of the future reward, the smaller Gamma means that the agent will consider only the immediate rewards as opposed to

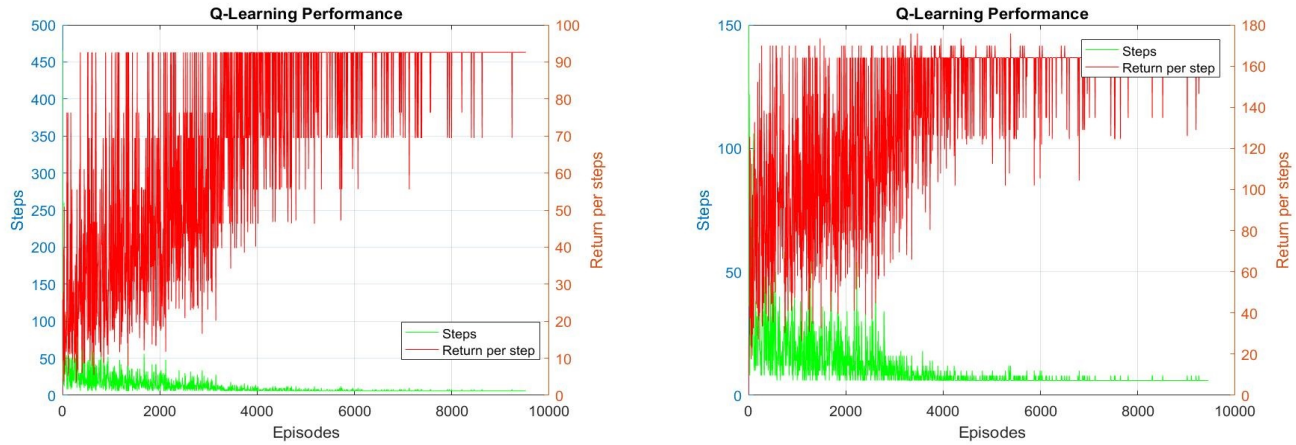

Figure 11: Left: Q-learning performance- Case 2. Gamma 0.10. Right: Gamma 0.50

higher Gamma value where the agent will consider both immediate and future reward. In base case 1, Gamma value was set to 0.8, the return per step was the highest in our experiment at approximately 300 (Figure 10). However, for Gamma value 0.1 and 0.5 the return per step severely dropped to 90 and 160 respectively with less fluctuation as Gamma decreases.Additionally, at 0.1 the agent has converged quicker at approximately 3300 episodes as opposed to 0.8 and 0.5 where the algorithm converged after 4000 episodes.Moreover, varying Gamma values 0.1, 0.5 and 0.8 did not have an effect on the agent's task in finding the shortest path. It was able to get to the goal state traversing the states shown in Figure 2 left. It was also noted that the values in the converged Q-matrix were smaller compared to base case 1. This also indicates that the agent was is learning slower in comparison to case 1. *Note Case 2: exploitCount =59121 exploreCount = 44841*

8 Case 3

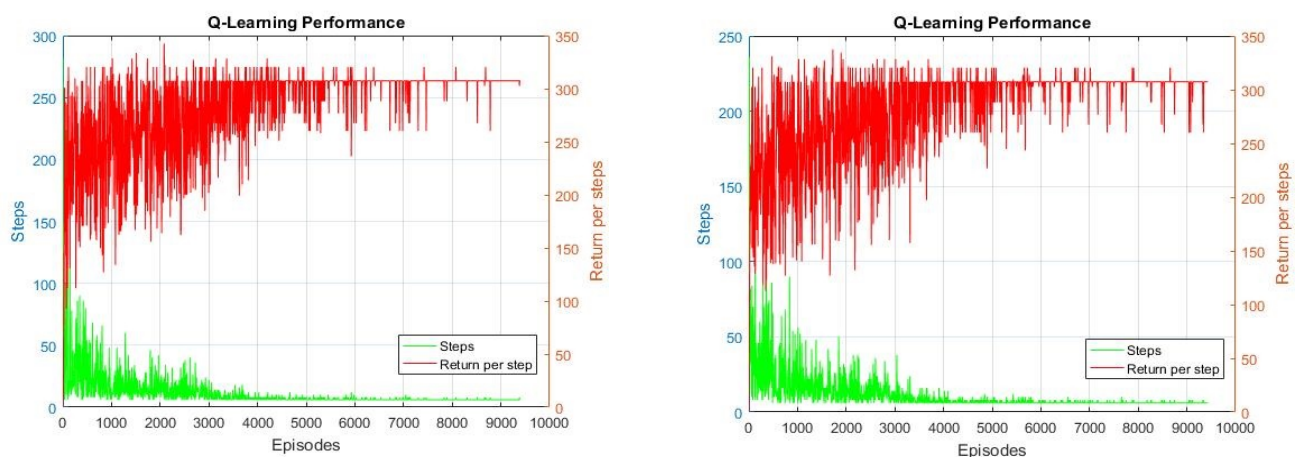

Figure 12: Left: Q-learning performance- Case 3. Alpha 0.20. Right: Alpha 0.50

Alpha is the learning rate, higher alpha value leads to quicker learning. It can be observed that changing alpha, did not lead to any major changes in the Figure 12. Both graphs in Figure 12 where alpha values set to 0.2 and 0.5 respectively, are similar to Figure 10 in base case 1 where alpha is 0.8. The return was approximately 300 in all cases. Moreover, the agent has converged and

reached an optimal asymptotic level after around 4000 episodes and the number of steps stabilised at approximately 6 steps similar to case 1 and case 2. Also, the converged Q-matrix is exactly the same base case 1.

It seems that because our environment is not dynamic (walls do not change), and all the paths are fixed, the alpha value does not change much in the results. In a deterministic environment, all next possible transition states remain the same. Therefore, changing alpha value will only change how much we will consider the new Q-value before adding it to the old Q-value.

9 Case 4

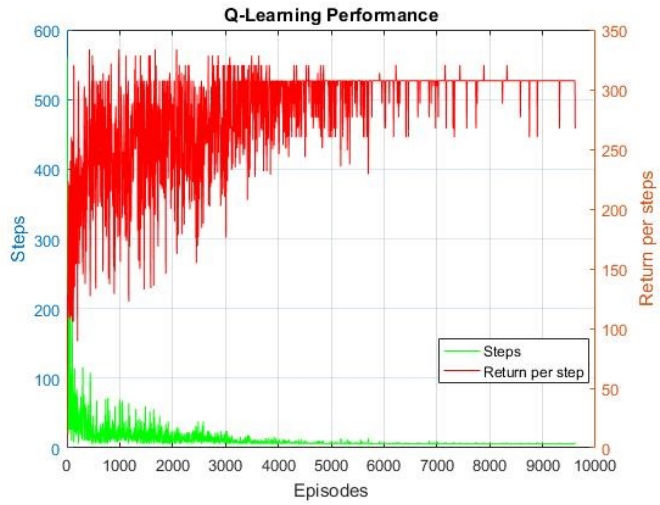

Figure 13: Q-learning performance- Case 4. Epsilon 1 (random policy)

We expected a slight variation in the results from base case 1 , since epsilon was set to 0.9. The expectation was to observe more randomness at the start in comparison to base case 1. The results proved that what we have hypothesised was correct. It took more steps and time initially as the agent was exploring all the paths to the goals state. Furthermore, it can be seen from Figure 13 and 10 that the algorithm converged after almost 4000 episodes. This is due to the decay factor that decreases the epsilon value after many episodes (shown in ϵ-greedy pseudocode above). Moreover, the agent was able to find the optimal path by reaching the goal state with only 6 steps.

Epsilon of value zero was also tested in our experiment,we did not include the resulting graph in this report since the agent could not reach its goal state. Because when epsilon is set to zero, the agent starts by exploiting without any explorations. If the agent does not start by exploring it would not be able to find the paths to the goal state which results in the agent getting trapped at the start state and does not know where to go from there.Therefore, it will not be able to find the possible steps or next states. And since the agent will not learn the paths, the Q-matrix will not be updated. Additionally, different values of epsilon were tested and it was noted that any value ϵ below 0.6 will not work.

10 Case 5

Two cases were attempted in this section

10.1 Case 5A slight changes in Reward and State

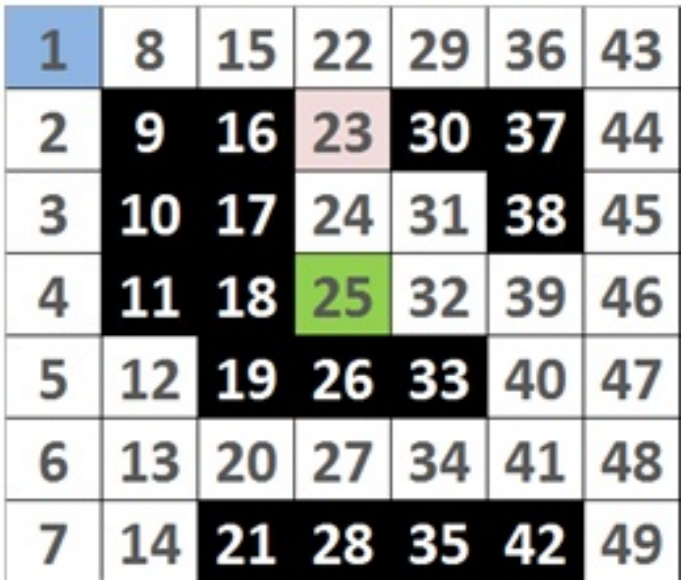

Figure 14: Penalised maze case 5. Blue is the initial start state, pink represents the obstacle and green is the goal state

For case 5A, the environment was slightly changed. In case 1, there were three paths that the agent can take in order to reach to the goal state as shown in Figure 2. The first path shown in Figure 2 left is the shortest path. For case 5, the shortest path has an obstacle at state 23 that will damage the robot's body if it choose to take this path (Figure 14). On one hand, if the robot takes the shortest path it be awarded 250 and penalised by -25 points (Figure 15). On the other hand, if it takes the medium or longest path presented in Figure 2 middle or right-most respectively, the robot will be awarded 1000.

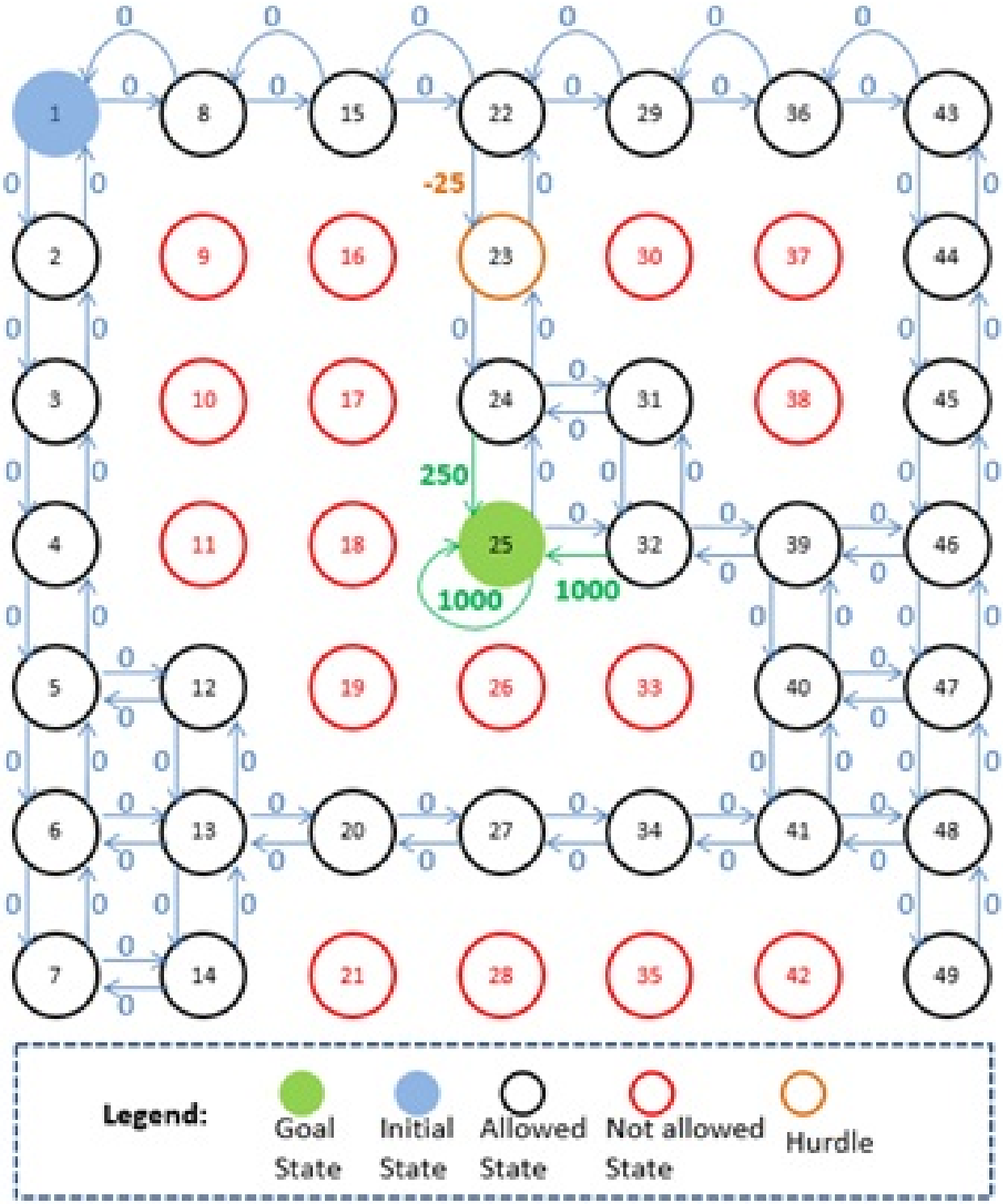

Figure 15: New representation of the domain with one penalised state

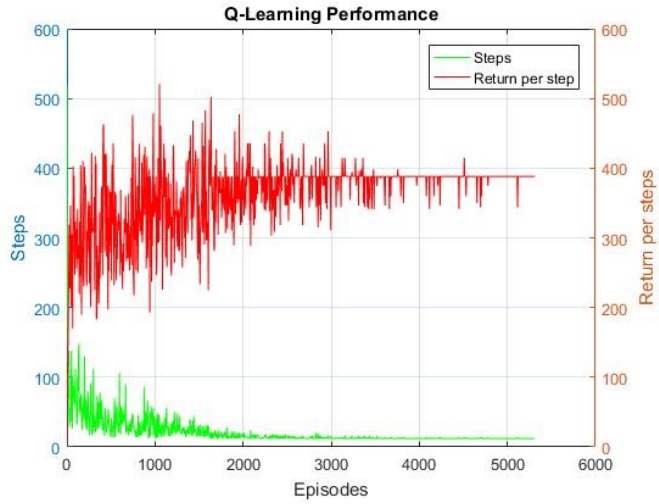

Figure 16: Q-learning performance- Case 5A

Figure 16 shows that the agent took more than 500 steps initially to reach the goal state since we are still using epsilon greedy similar to base case 1. Epsilon was high at the start which means the agent was exploring, resulting in high fluctuation before 3000 episodes (red lines). It is shown that the red line became more stable and less spiky as the number of episodes increased and the agent exploits more. However, unlike base case 1, the algorithm converges quicker reaching an asymptotic level at approximately 1800 episodes and the return was also higher at almost 390 (since the reward was set to 1000). Also, it was interesting to notice that the agent did not take the shortest path where it would have been penalised. The agent however preferred the medium path (Figure 2 middle) avoiding the penalty at state 23. The optimal number of steps reached in this case was 12. The route the agent took is: 1,8, 15,22,29,36,43,44,45,46,39,32,25.

10.2 Case 5B extreme changes in Reward and State

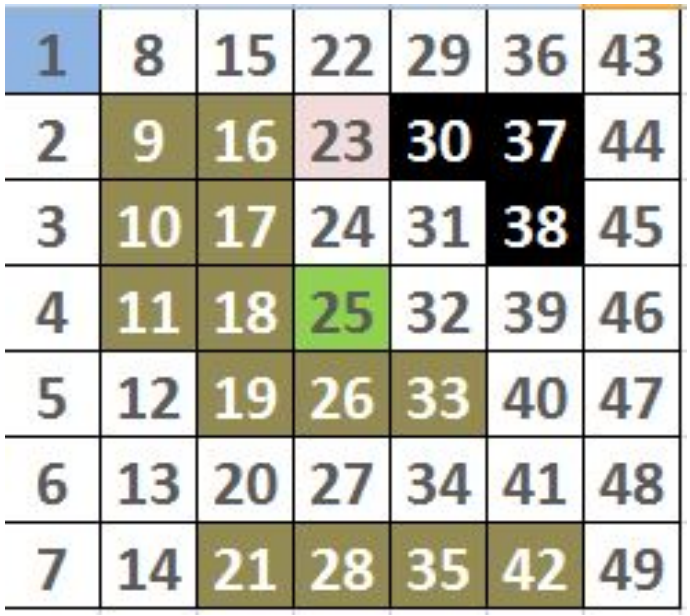

Figure 17: Case 5B agent is allowed to enter pale green wall with a penalty of -5, black walls are not allowed and green wall has higher penalty of -25. Reward of reaching final goal state-25 is 1000 points

In order to study the effect of changing the state further, intensive changes were made in the state. This time an agent is allowed to enter into some wall states with a penalty of -5 points shown in pale green in the Figure 17.
In case 5B wall states 9,10,11,16,17,18,19,26,33,21,28,35 and 42 were allowed to enter with a penalty of -5 points. However, certain wall remain intact as in Case 1 and agent cannot enter into 30,37 and 38 respectively. We have kept state 23 as a special case with a penalty of -25, exactly as it was introduced in Case 5A. Graph diagram is shown in Figure 18

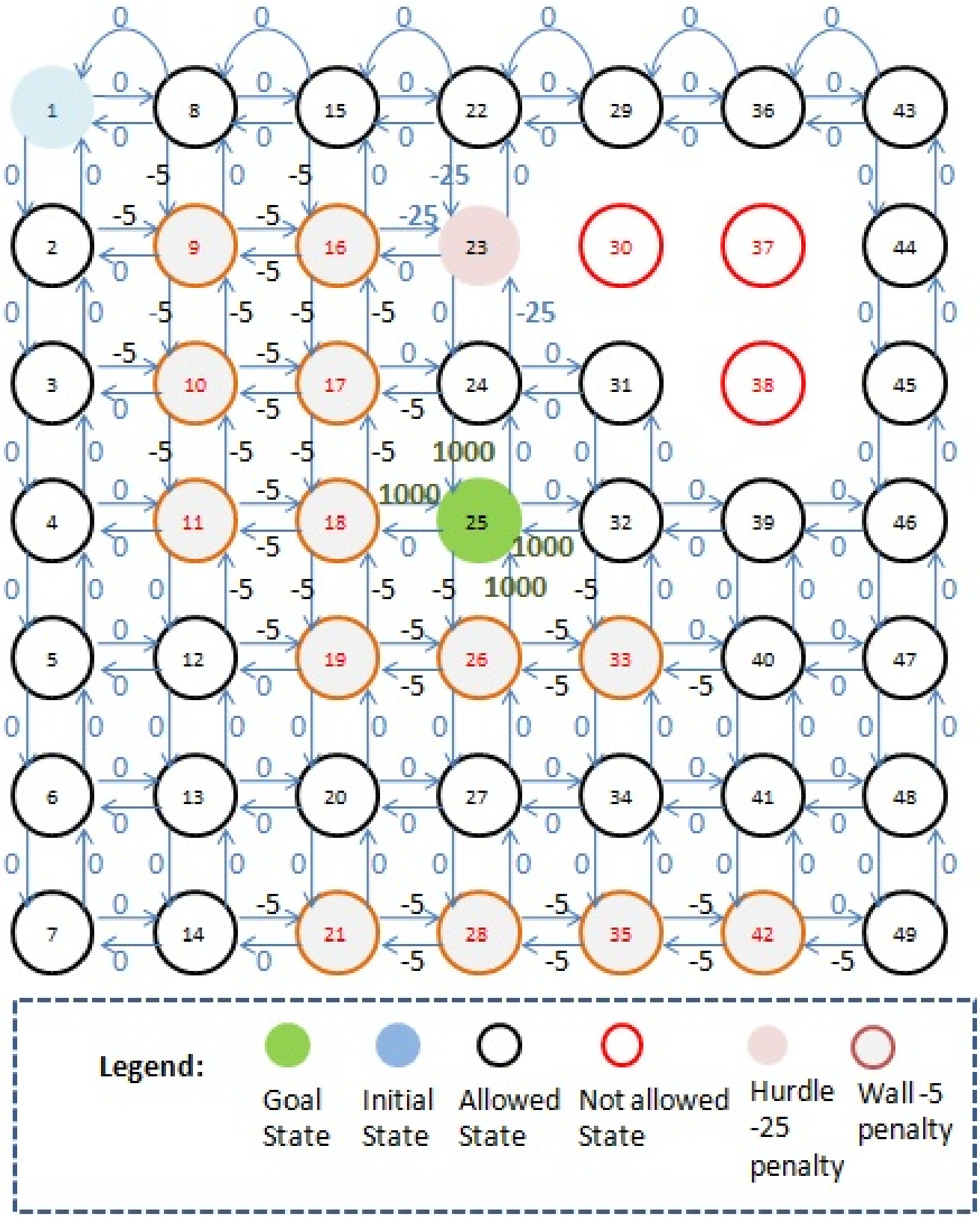

Figure 18: Graph with updated rewards for Case 5B

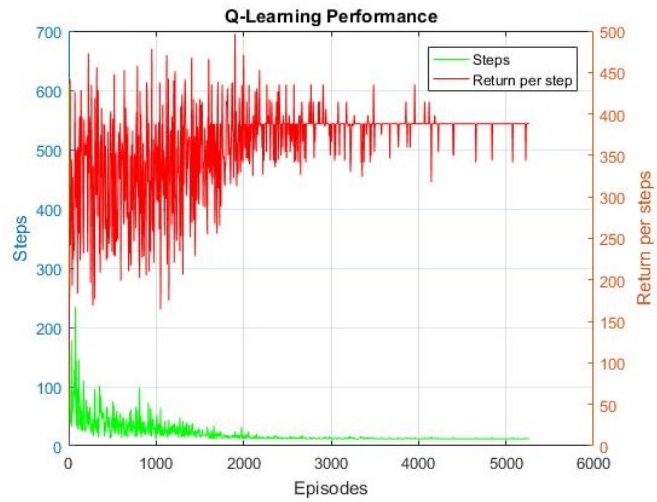

Figure 19: Graph with updated rewards for Case 5B

Figure 19 shows that the agent took more than 600 steps initially for completing one episode, roughly hundred steps more than Case 5A. Both graphs (Figure 16 and 19) in case 5A and 5B are very similar. The agent preferred the medium path as shown in Figure 2-middle just like the case in 5A. Also, it converged after approximately 1800 steps and 390 return per step. Figure 19 however, showed more fluctuation compared to Figure 16, before 2000 episodes. This can be explained by the fact that the agent can now traverse through walls and the maze contain more penalised states. Therefore, it took the agent more steps to learn its way around it and reach the goal state. Moreover, increasing the reward to a 1000 points and allowing the agent to enter the wall state with a low penalty of -5 did not encourage the agent to converge to a route with less number of steps. We can safely conclude that agent converged at the safest route avoiding any sort of penalty even if it involves more number of steps to reach the goal state. The route taken by agent is 1,8, 15,22,29,36,43,44,45,46,39,32,25. Q matrix is also same for both Case 5A and Case 5B.

Note Case 5B:exploit Count =65023 explore Count = 44937

11 Conclusion

In base case 1, Figure 10, alpha and gamma values were set to 0.8 and epsilon set to 0.9. We observed from the graph that initially the agent was exploring hence the high number of steps taken and the high fluctuation in the return per step. As the agent is learning, it started to exploit after certain number of episodes and converged after approximately 4000 episodes with a cumulative reward 300.

In case 2, we tested gamma value with 0.1 and 0.5, Figure 11 Left and right respectively. In comparison with case 1, it can be seen that as gamma decreases, the number of steps decreased (less fluctuation in the green line and evened out faster) and the return per step severely dropped with more fluctuation in the red lines. Gamma did not only lower the return but it affected the convergence of the return per step, it takes more episodes to stabilise the return as gamma decreases. In addition, gamma value influence the Q-matrix value. The Q-matrix values were smaller than base case 1, indicating that the agent is learning at slower rate. Therefore, since in our case the rewards between the transition states are set to zero (expect for the goal state), gamma value should be set higher so that the agent gives the correct approximation of the Q-values while traversing to the goal

state.

Moving on to the third case, we tested lower alpha values (Figure 12 left alpha 0.2 and right alpha 0.5). Ideally, higher alpha value means that the matrix should converge quicker. The Q-matrix values are exactly the same as in case 1, which also means that it is higher than case 2. The only difference was the slight variation in the convergence. In comparison to base case 1 where alpha was the highest, it converged quicker than case 3 (lower alpha led to more episodes to converge). It was also noted in our experiment, for higher alpha values it takes the code more episodes to finish/stop. However, in comparison to case 2, when gamma value was set to 0.1 it converged even quicker.

As for case 4, we tested with epsilon 1 and reported on our observations when for epsilon values below 0.6. From Figure 13, we can see that the graph is less fluctuated in general compared to all other cases. The return per step was approximately 300 similar to case 1 and 3. However, the number of steps taken initially is the highest amongst all other cases since with higher epsilon the agent tends to explore more. And this case also showed the quickest convergence (amongst the first four cases) in term of number of episodes which was reported at approximately 3000 episodes. We also reported that when epsilon is lower than 0.6 the agent will not be able to leave the start state and will not find its way to the goal state. This is due to the fact that the agent needs to explore the paths first before exploiting.

In case 5A and 5B, we used the same set of parameter values as in base case 1. Although in case 5B the environment was dynamic, the graphs (Figure 16 and 19) are almost exactly the same in terms of number of steps, return, convergence, and Q-matrix values. The only difference was seen in the higher number of steps taken initially in case 5B. In comparison to all other cases, case 5A and B, the agent explored more initially (took more steps), the return was the highest, and the agent converged quicker at approximately 1800 episodes. However, the Q-marix values were lower than case 1 and case 3 but higher than case 2 where gamma was 0.5 and 0.1.

References

[1] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction. 1998," *A Bradford Book*, 1998.

[2] M. Dorigo and H. Bersini, "A comparison of q-learning and classifier systems," *From animals to animats*, vol. 3, pp. 248–255, 1994.

[3] "Kardi Teknomo q-learning by examples." `http://people.revoledu.com/kardi/tutorial/ReinforcementLearning/index.htm`. Accessed: 2017-02-30.

[4] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[5] S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári, "Convergence results for single-step on-policy reinforcement-learning algorithms," *Machine Learning*, vol. 38, no. 3, pp. 287–308, 2000.