Artificial Intelligence Assignment 4 Report

Pradeep Kumar 2019CSM1008

17 May 2020

1 Problem Statement

In this assignment, I have to implement solver to solve Cliff walking problem using two different techniques of Reinforcement Learning. One is Q-Learning and second is State–Action–Reward–State–Action (SARSA).

2 Introduction

Reinforcement learning is a computational approach to learning whereby an agent tries to maximize the total amount of reward it receives when interacting with a complex, uncertain environment. Consider the grid-world of Cliff walking problem shown below:

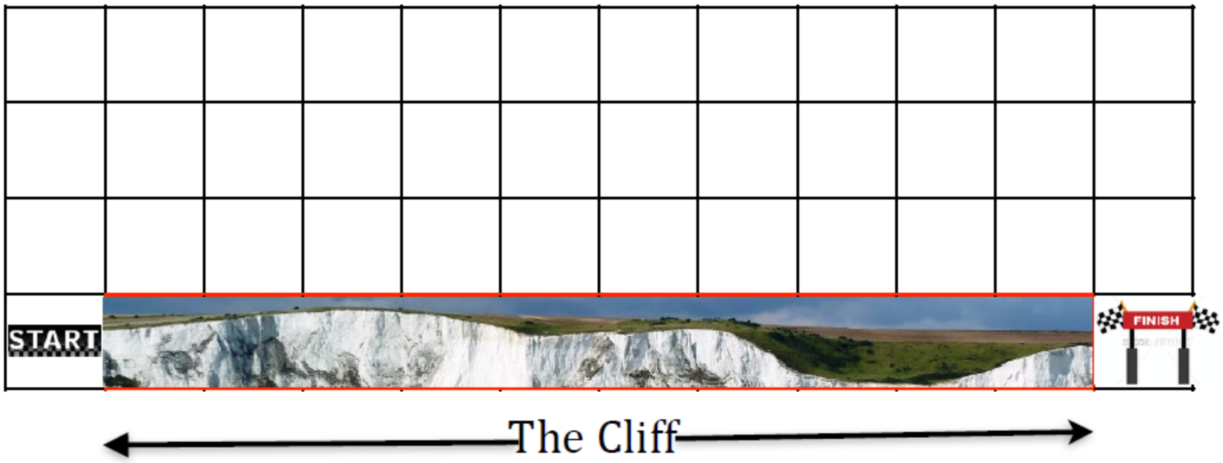


Figure 1: Environment of Cliff walking problem

2.1 Q-Learning

Q-Learning is a basic form of Reinforcement Learning which uses Q-values (also called action values) to iteratively improve the behavior of the learning agent.

Given a set of states and actions, Q-Learning algorithm is based on continuous update (learning) of a function Q. For each taken action agent receives a reward (or penalty) and updates the Q function. The goal of the agent is to maximise the reward.

$$\begin{aligned} Q[s,a] &= Q[s,a] + \alpha * (R + \gamma * Max[Q(s',a')] - Q[s,a]) \\ Q[s,a] &= (1-\alpha) * Q[s,a] + \alpha * (R + \gamma * Max[Q(s',a')]) \end{aligned} \tag{1}$$

where :

- Q : Value function for each state-action pair.
- s : Any state from environment.
- a : Any action that can be taken from a given state.
- s' : Next state after performing action a at state s.
- a' : Next state action
- α : The learning rate, indicates how much new calculated information is important compared to older information. It's between 0 and 1 when 0 means agent doesn't learn and when 1 means agent ignores what it already learned
- γ : Discount Factor, determines how much future rewards are important, when close to 0, agent gives more considerations to current rewards, when close to 1, agent considers more future rewards when updating Q.

2.2 State–Action–Reward–State–Action (SARSA)

This algorithm is almost similar like Q-Learning but it's updating the Q-value depends on the current state Q-values as well as next state Q-values. The goal of the agent is to maximise the reward.

$$Q[s,a] = Q[s,a] + \alpha * (R + \gamma * Q(s',a') - Q[s,a]) \tag{2}$$

3 Given Problem Parameter Description

3.1 Policy

I am using ϵ-greedy policy for choosing the action from all possible actions. It's using the current Q-value estimations. It goes as follows :

- With ϵ probability choose the action which has the highest Q-value.
- With $1-\epsilon$ probability choose any action at random.

3.2 Reward

I am making some assumptions for reward like:

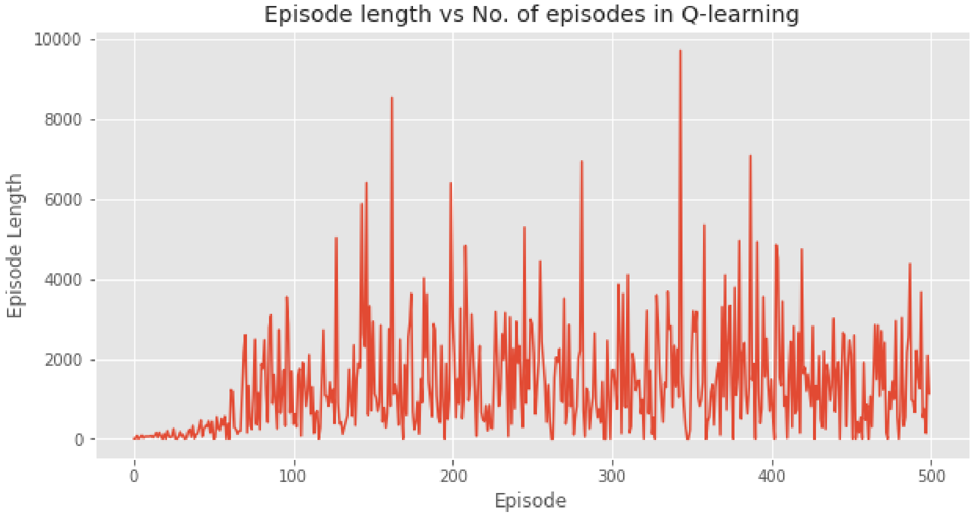
- Reward of each position of cliff: -100
- Reward of start position: -20
- Reward of goal position: +100
- otherwise: -1

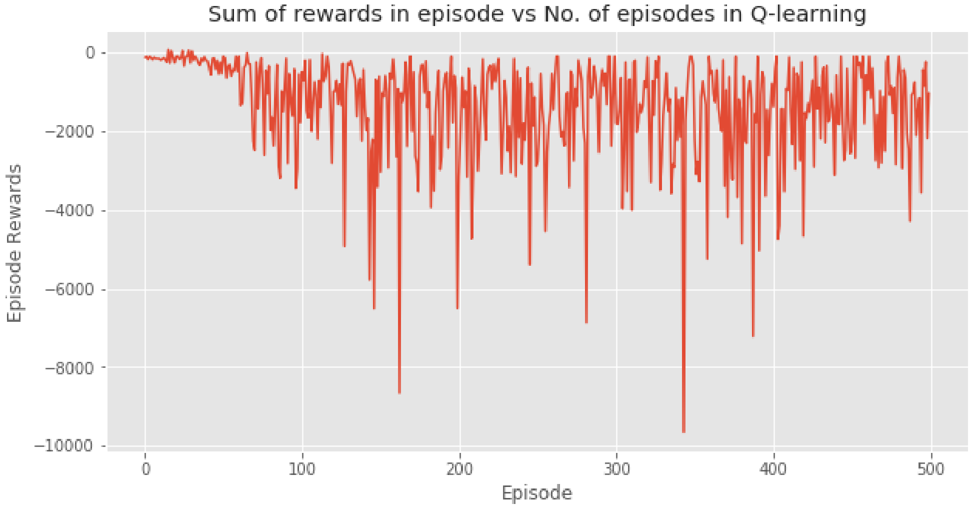
3.3 Action

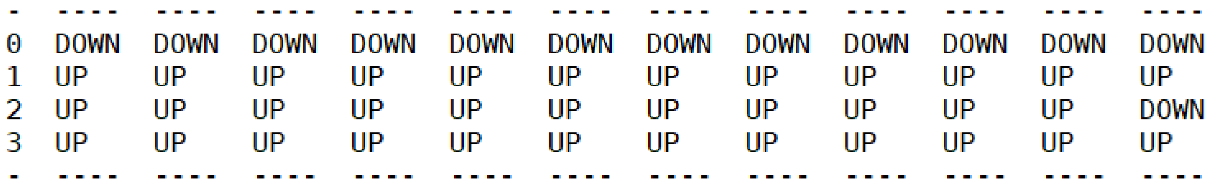
There are only 4 actions: Left, Right, Up and Down.

4 Experimental Results

4.1 Case-1: $\alpha = 0.5$, $\gamma = 0$, $\epsilon = 0.1$, episodes = 1000

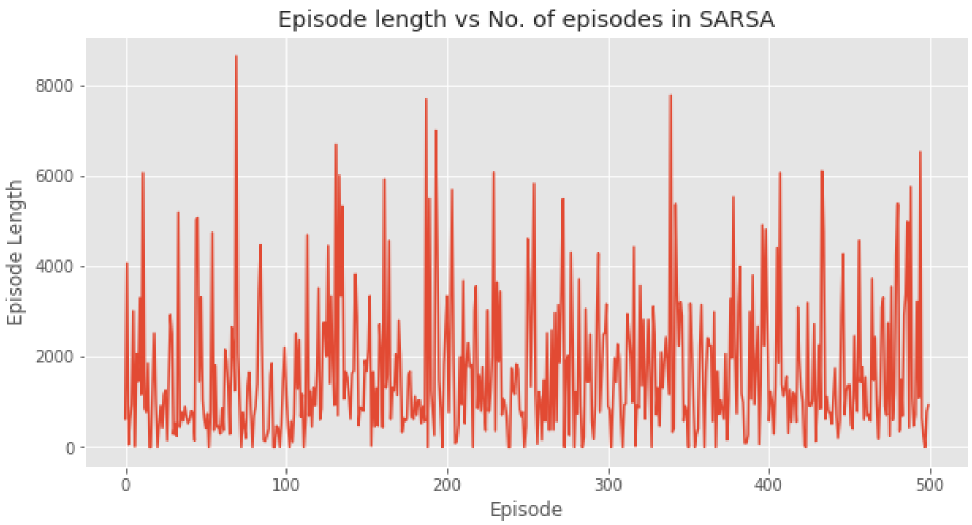


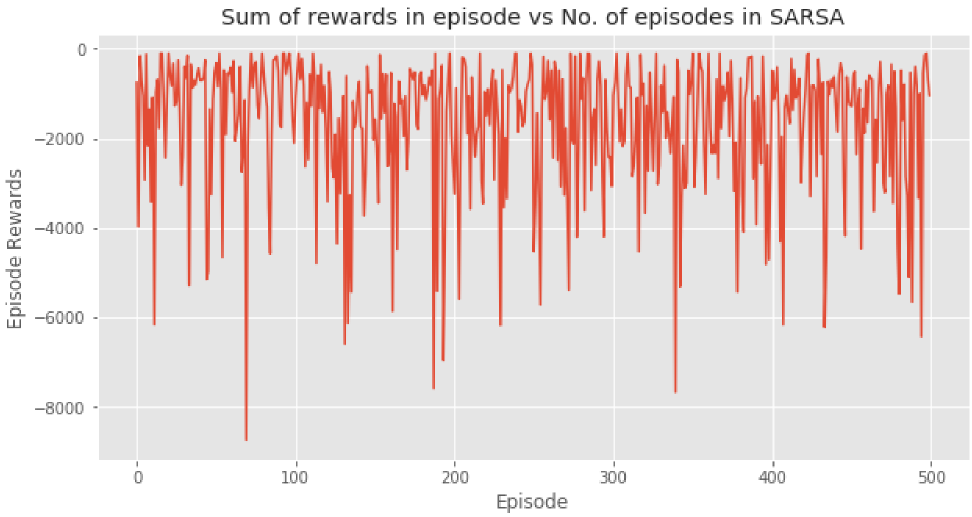


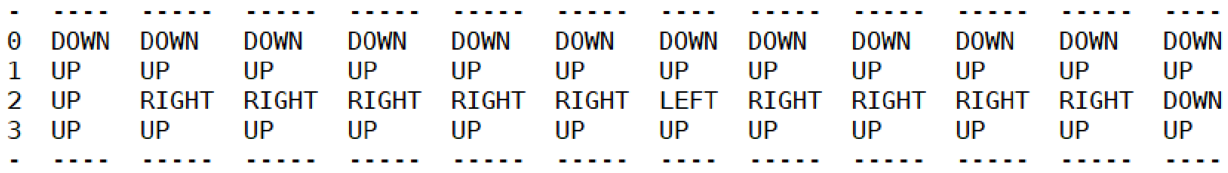


```
-  ----  ----  ----  ----  ----  ----  ----  ----  ----  ----  ----  ----
0  DOWN  DOWN  DOWN  DOWN  DOWN  DOWN  DOWN  DOWN  DOWN  DOWN  DOWN  DOWN
1  UP    UP    UP    UP    UP    UP    UP    UP    UP    UP    UP    UP
2  UP    UP    UP    UP    UP    UP    UP    UP    UP    UP    UP    DOWN
3  UP    UP    UP    UP    UP    UP    UP    UP    UP    UP    UP    UP
-  ----  ----  ----  ----  ----  ----  ----  ----  ----  ----  ----  ----
```

Figure 2: Optimal Policy of Q-Learning algorithm







```
-  ----  -----  -----  -----  -----  -----  ----  -----  -----  -----  -----  ----
0  DOWN  DOWN   DOWN   DOWN   DOWN   DOWN   DOWN  DOWN   DOWN   DOWN   DOWN   DOWN
1  UP    UP     UP     UP     UP     UP     UP    UP     UP     UP     UP     UP
2  UP    RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  LEFT  RIGHT  RIGHT  RIGHT  RIGHT  DOWN
3  UP    UP     UP     UP     UP     UP     UP    UP     UP     UP     UP     UP
-  ----  -----  -----  -----  -----  -----  ----  -----  -----  -----  -----  ----
```

Figure 3: Optimal Policy of SARSA algorithm

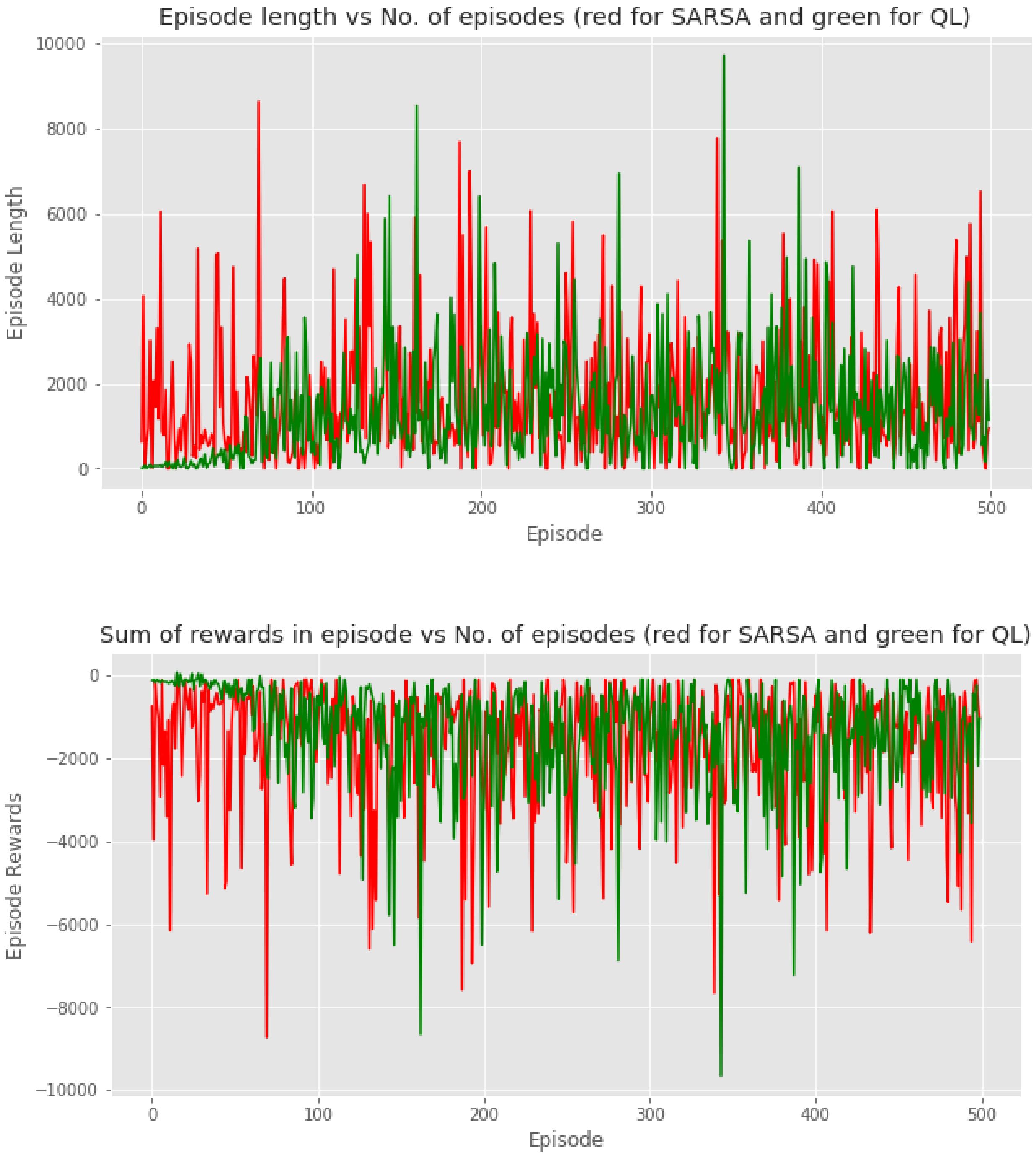
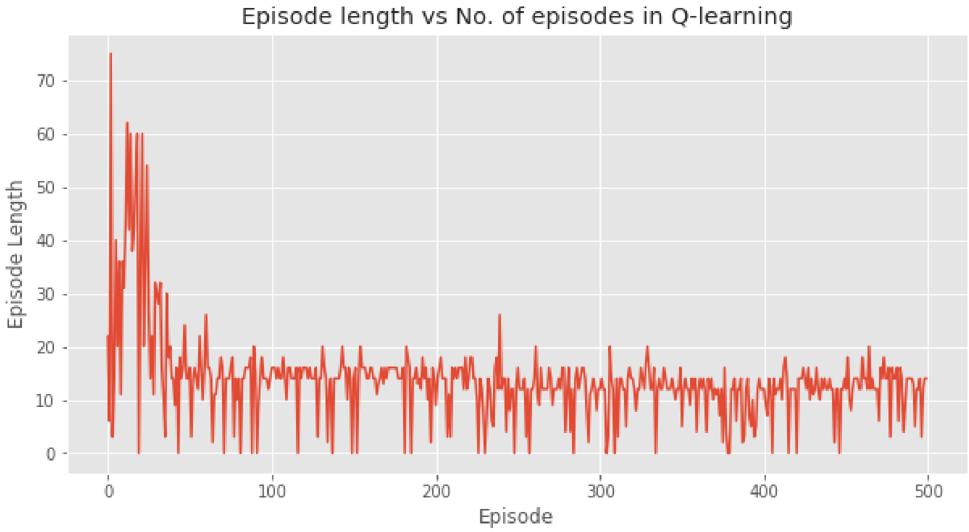
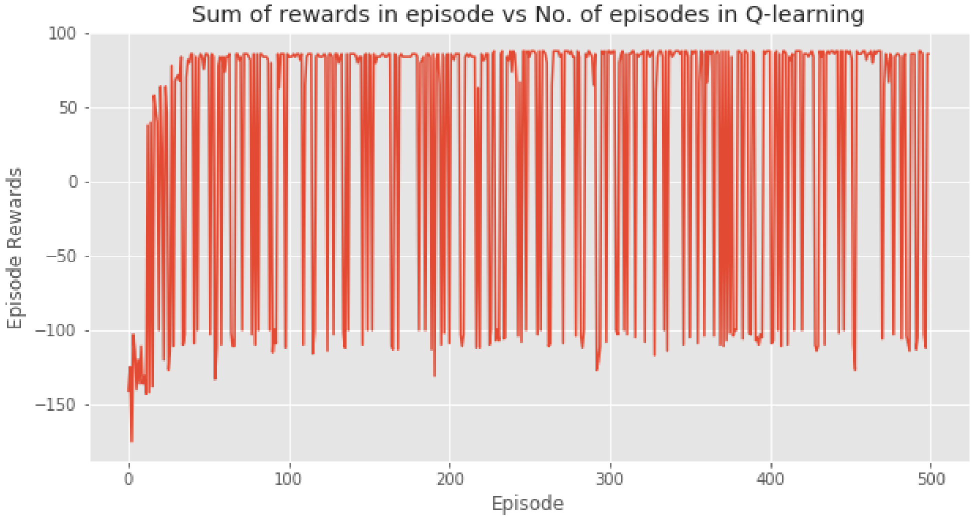
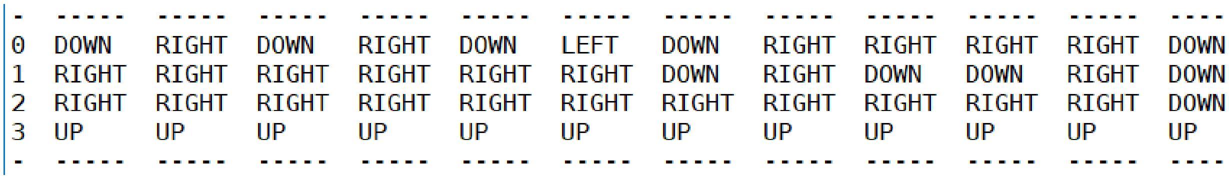


Figure 4: Comparison between Q-learning and SARSA algorithm

4.2 Case-2: $\alpha = 0.5$, $\gamma = 0.9$, $\epsilon = 0.1$, episodes = 1000

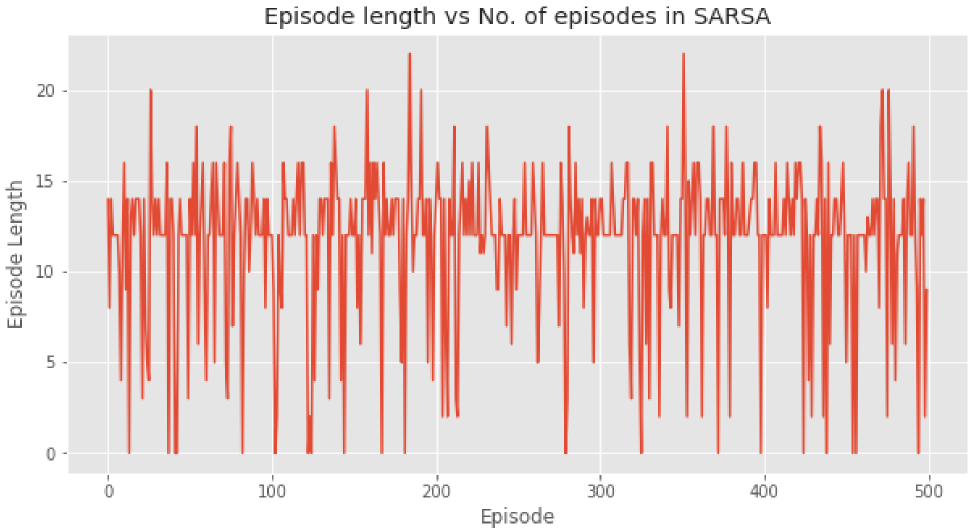


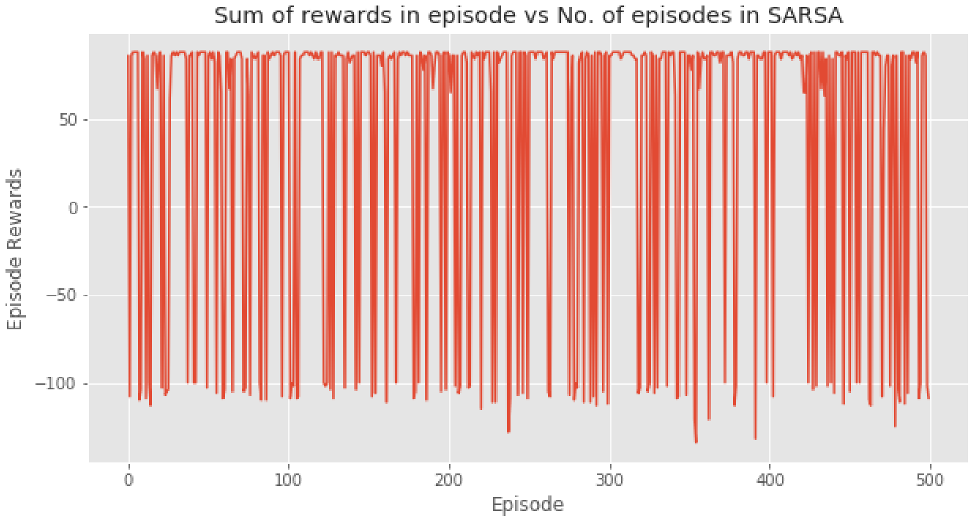




```
-  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  ----
0  DOWN   RIGHT  DOWN   RIGHT  DOWN   LEFT   DOWN   RIGHT  RIGHT  RIGHT  RIGHT  DOWN
1  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  DOWN   RIGHT  DOWN   DOWN   RIGHT  DOWN
2  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  DOWN
3  UP     UP     UP     UP     UP     UP     UP     UP     UP     UP     UP     UP
-  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  ----
```

Figure 5: Optimal Policy of Q-Learning algorithm





```
-  -----  -----  -----  -----  -----  -----  ----  -----  ----  ----  -----  ----
0  DOWN   DOWN   DOWN   RIGHT  DOWN   DOWN   DOWN  DOWN   DOWN  DOWN  DOWN   DOWN
1  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  DOWN  RIGHT  DOWN  UP    RIGHT  DOWN
2  UP     UP     RIGHT  RIGHT  RIGHT  RIGHT  UP    LEFT   UP    LEFT  RIGHT  DOWN
3  UP     UP     UP     UP     UP     UP     UP    UP     UP    UP    UP     UP
-  -----  -----  -----  -----  -----  -----  ----  -----  ----  ----  -----  ----
```

Figure 6: Optimal Policy of SARSA algorithm

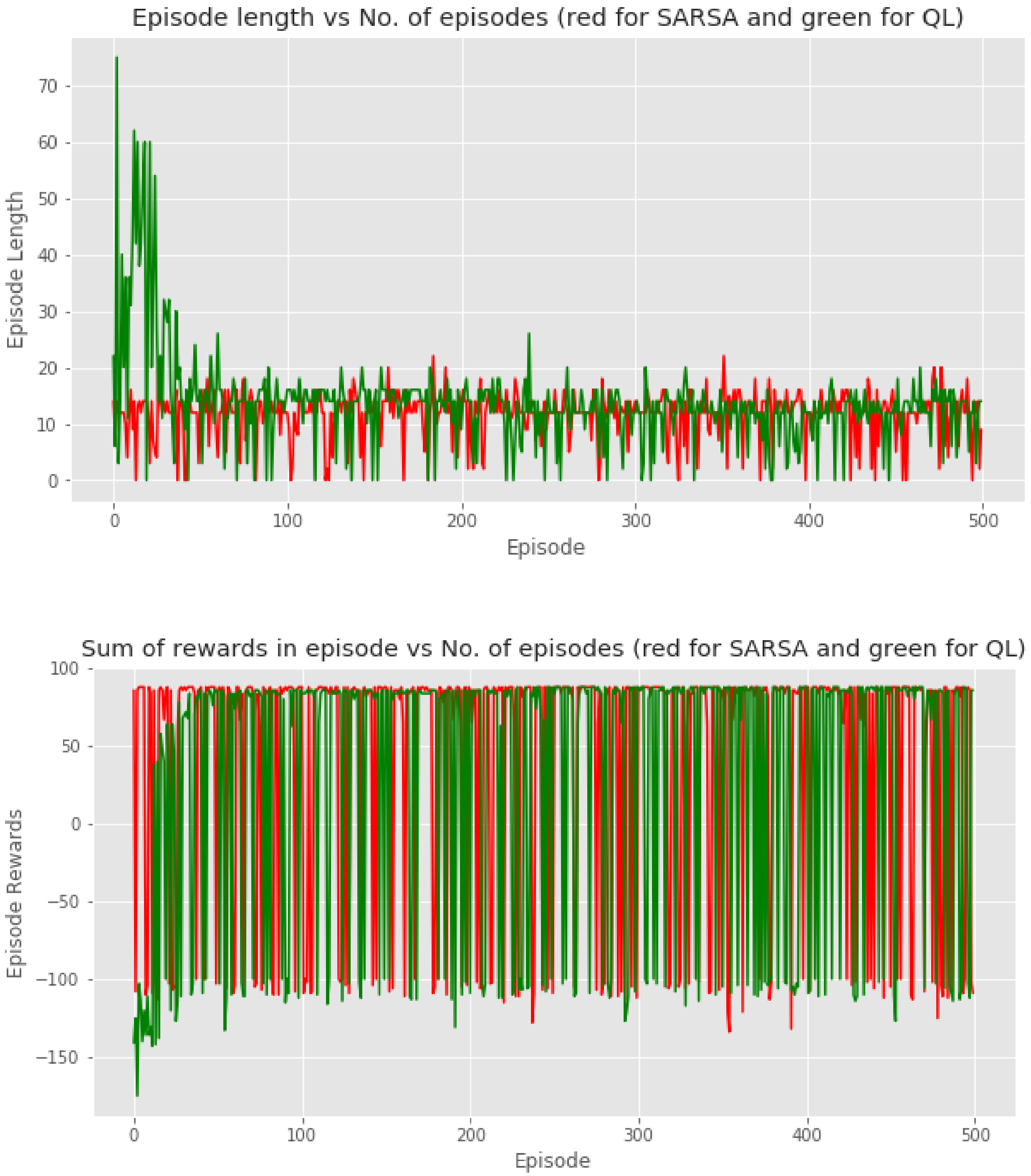
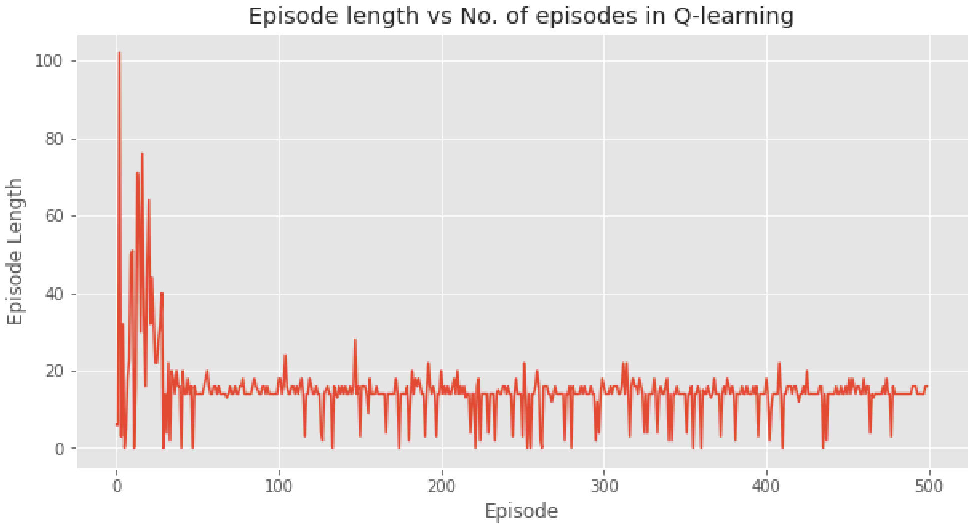
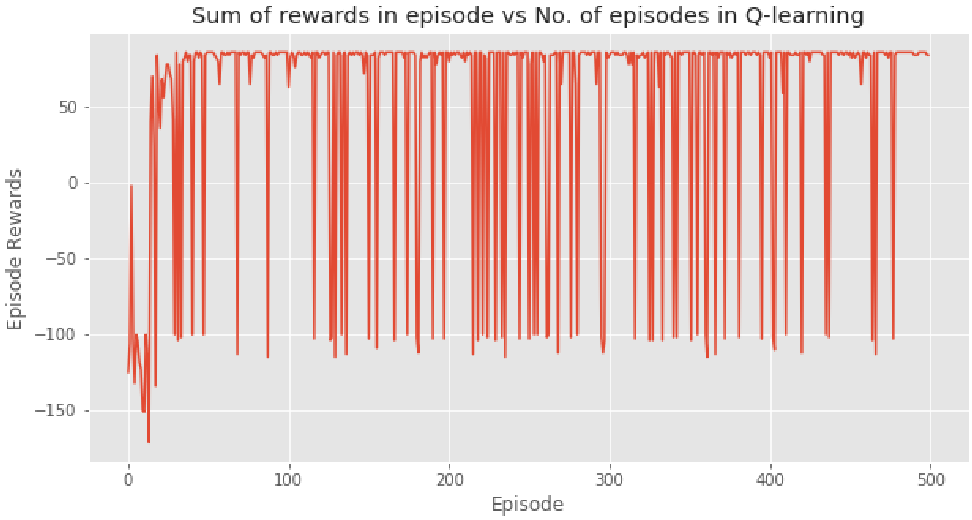
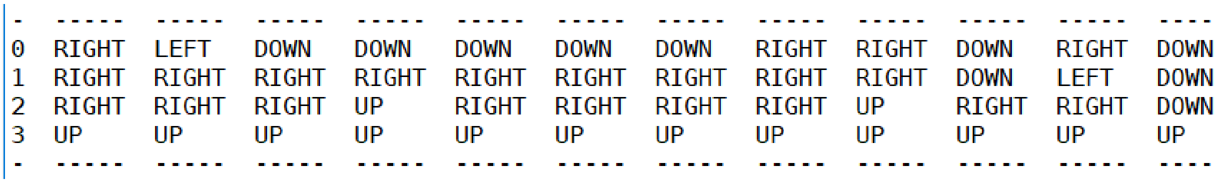


Figure 7: Comparison between Q-learning and SARSA algorithm

4.3 Case-3: α = 0.5, γ = 0.9, ϵ = Decreased to 0 from 0.1 with time, episodes = 1000

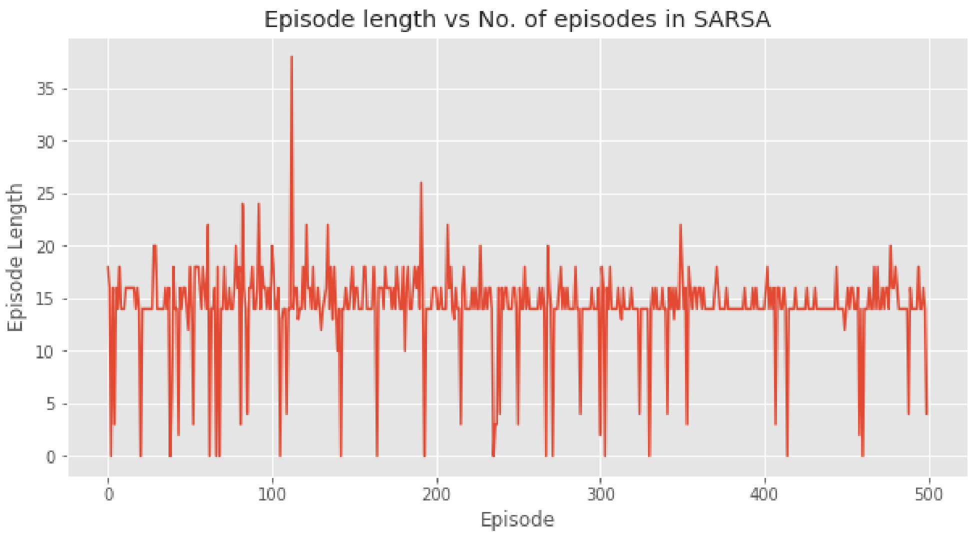


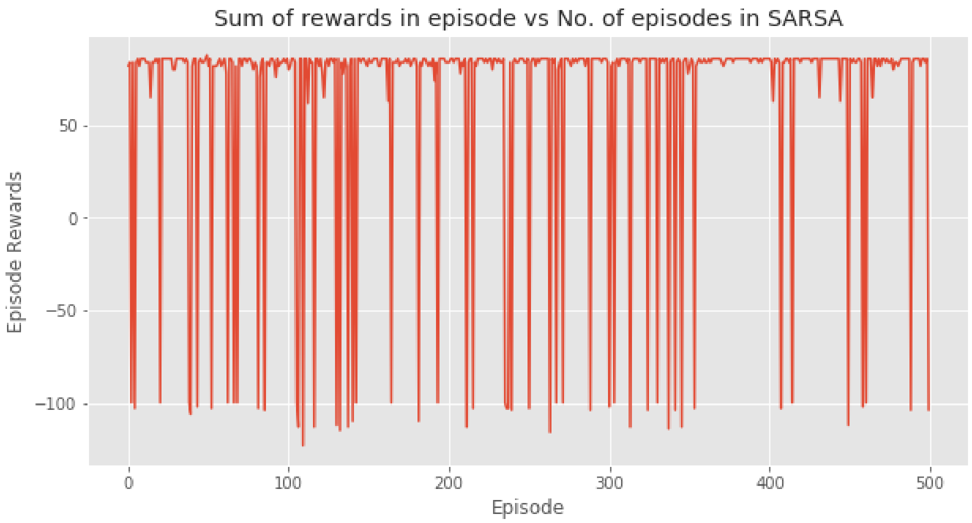


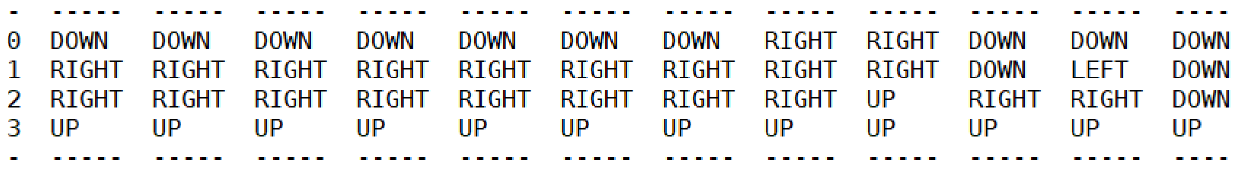


```
-  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  ----
0  RIGHT  LEFT   DOWN   DOWN   DOWN   DOWN   DOWN   RIGHT  RIGHT  DOWN   RIGHT  DOWN
1  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  DOWN   LEFT   DOWN
2  RIGHT  RIGHT  RIGHT  UP     RIGHT  RIGHT  RIGHT  RIGHT  UP     RIGHT  RIGHT  DOWN
3  UP     UP     UP     UP     UP     UP     UP     UP     UP     UP     UP     UP
-  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  ----
```

Figure 8: Optimal Policy of Q-Learning algorithm

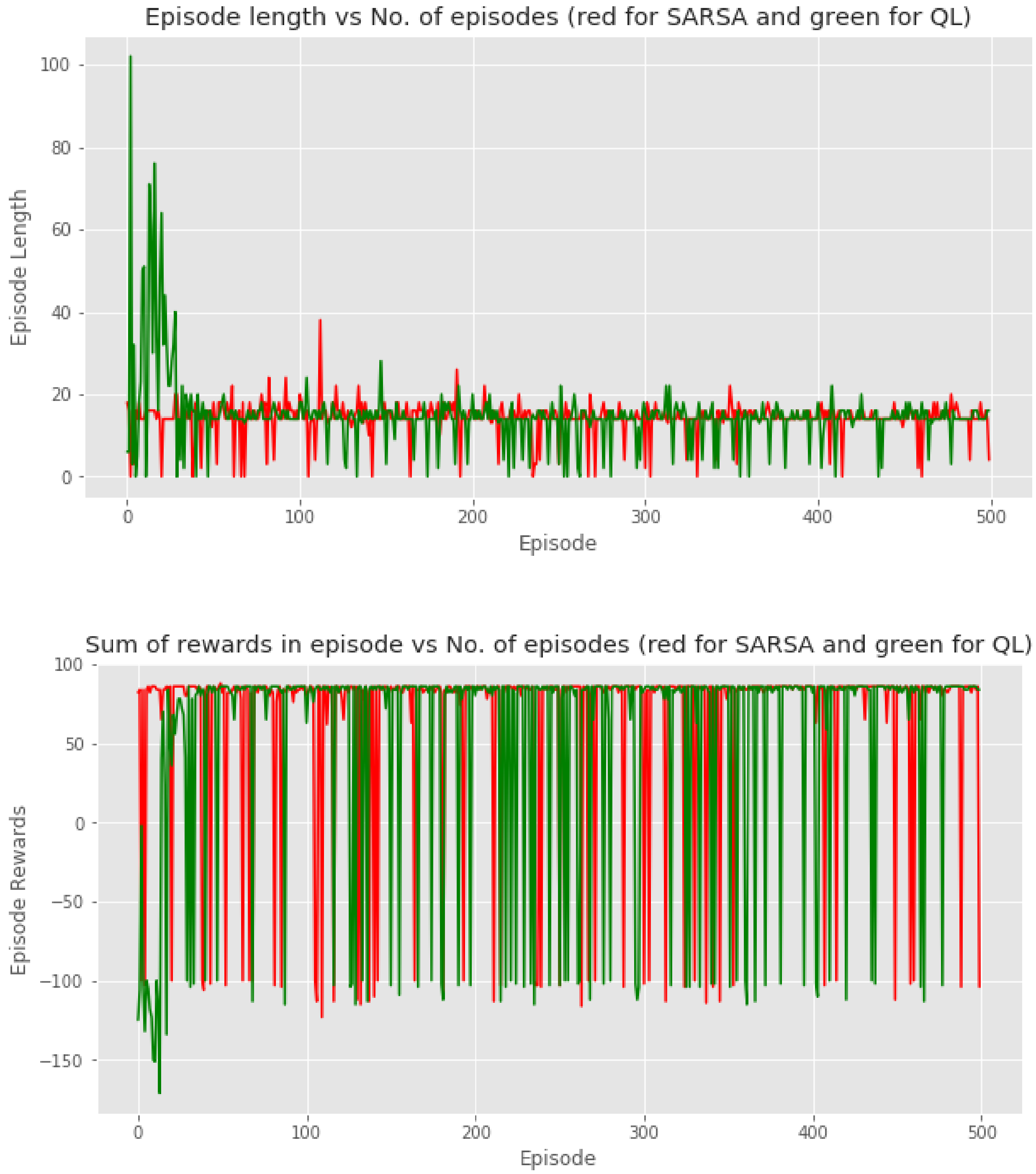






```
-  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  ----
0  DOWN   DOWN   DOWN   DOWN   DOWN   DOWN   DOWN   RIGHT  RIGHT  DOWN   DOWN   DOWN
1  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  DOWN   LEFT   DOWN
2  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  UP     RIGHT  RIGHT  DOWN
3  UP     UP     UP     UP     UP     UP     UP     UP     UP     UP     UP     UP
-  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  ----
```

Figure 9: Optimal Policy of SARSA algorithm

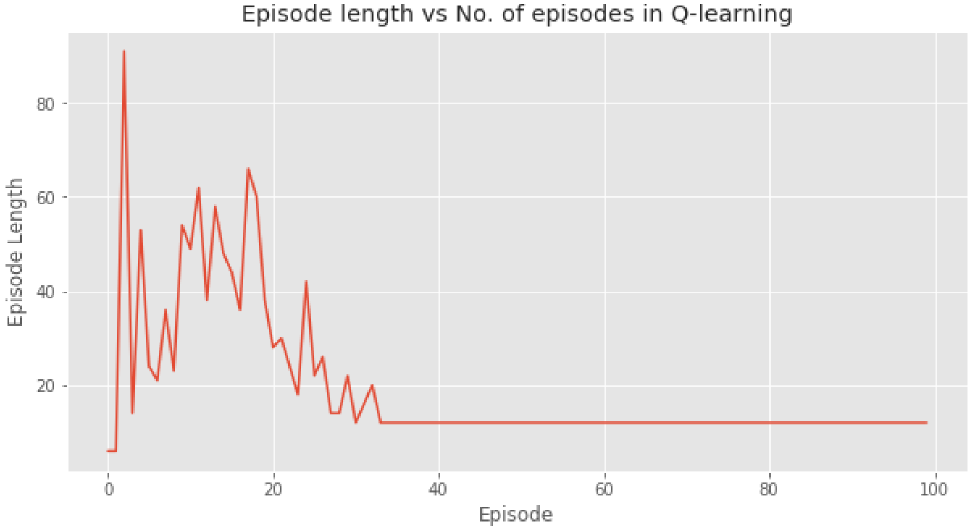


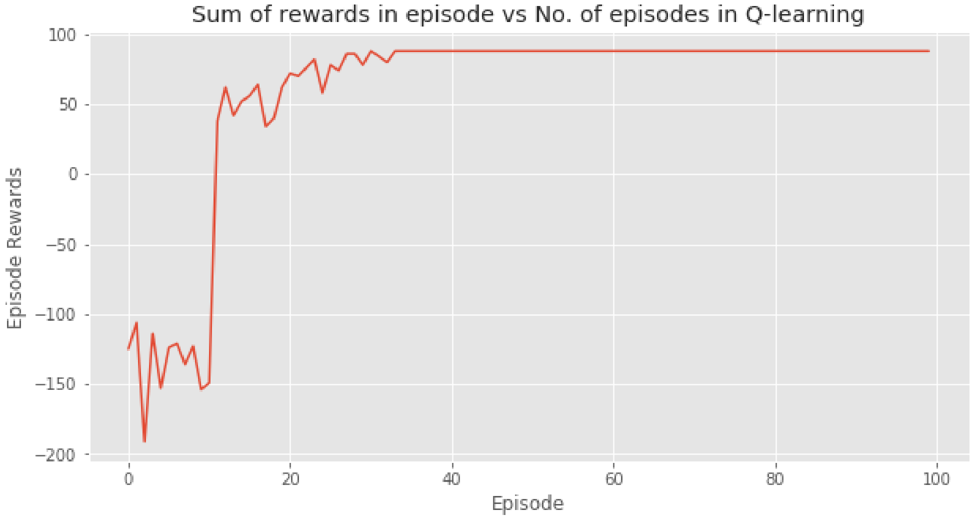
Episode length vs No. of episodes (red for SARSA and green for QL)

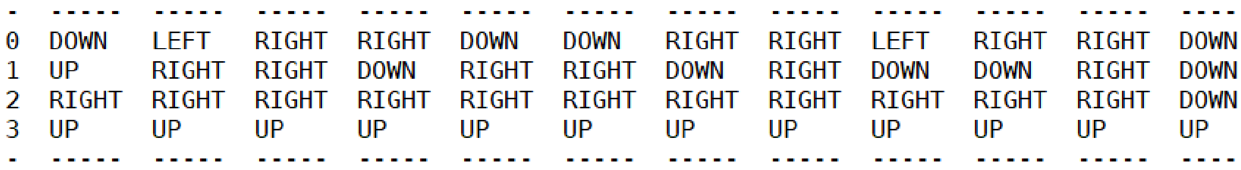
Sum of rewards in episode vs No. of episodes (red for SARSA and green for QL)

Figure 10: Comparison between Q-learning and SARSA algorithm

4.4 Case-4: When action selection is greedy and $\alpha = 0.5$, $\gamma = 0.9$, episodes = 1000

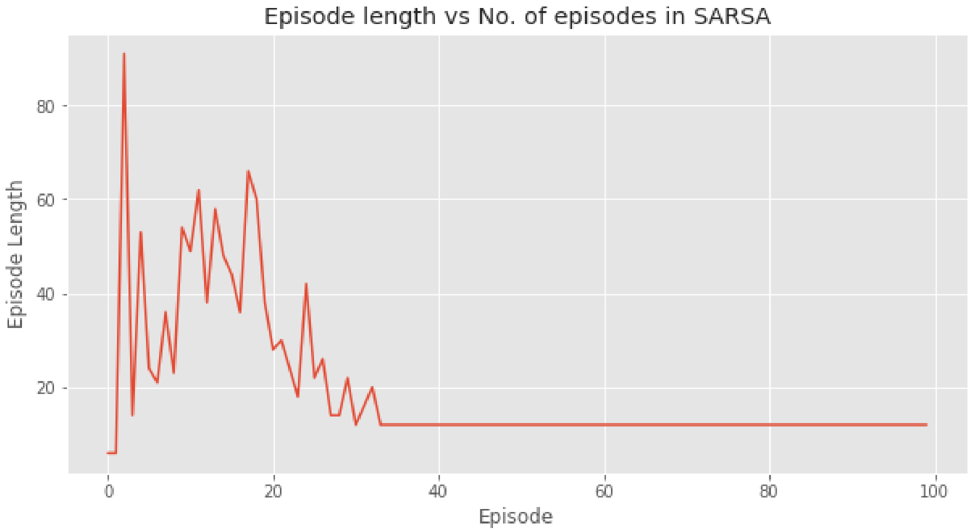


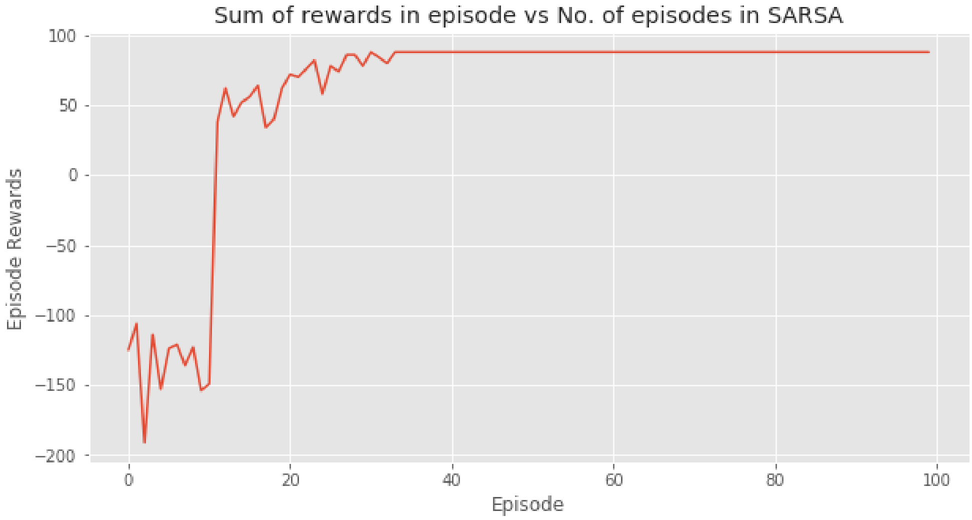




```
-  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  ----
0  DOWN   LEFT   RIGHT  RIGHT  DOWN   DOWN   RIGHT  RIGHT  LEFT   RIGHT  RIGHT  DOWN
1  UP     RIGHT  RIGHT  DOWN   RIGHT  RIGHT  DOWN   RIGHT  DOWN   DOWN   RIGHT  DOWN
2  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  DOWN
3  UP     UP     UP     UP     UP     UP     UP     UP     UP     UP     UP     UP
-  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  ----
```

Figure 11: Optimal Policy of Q-Learning algorithm





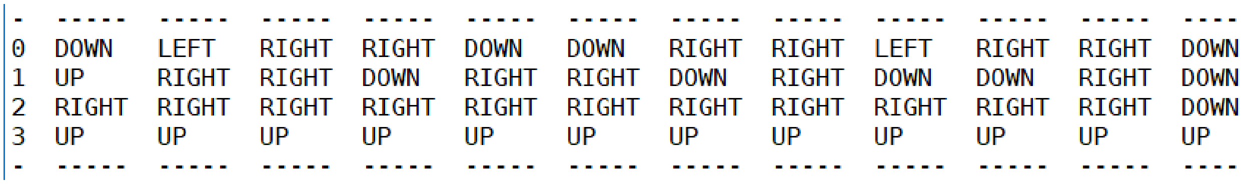

```
-  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  ----
0  DOWN   LEFT   RIGHT  RIGHT  DOWN   DOWN   RIGHT  RIGHT  LEFT   RIGHT  RIGHT  DOWN
1  UP     RIGHT  RIGHT  DOWN   RIGHT  RIGHT  DOWN   RIGHT  DOWN   DOWN   RIGHT  DOWN
2  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  RIGHT  DOWN
3  UP     UP     UP     UP     UP     UP     UP     UP     UP     UP     UP     UP
-  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  ----
```

Figure 12: Optimal Policy of SARSA algorithm

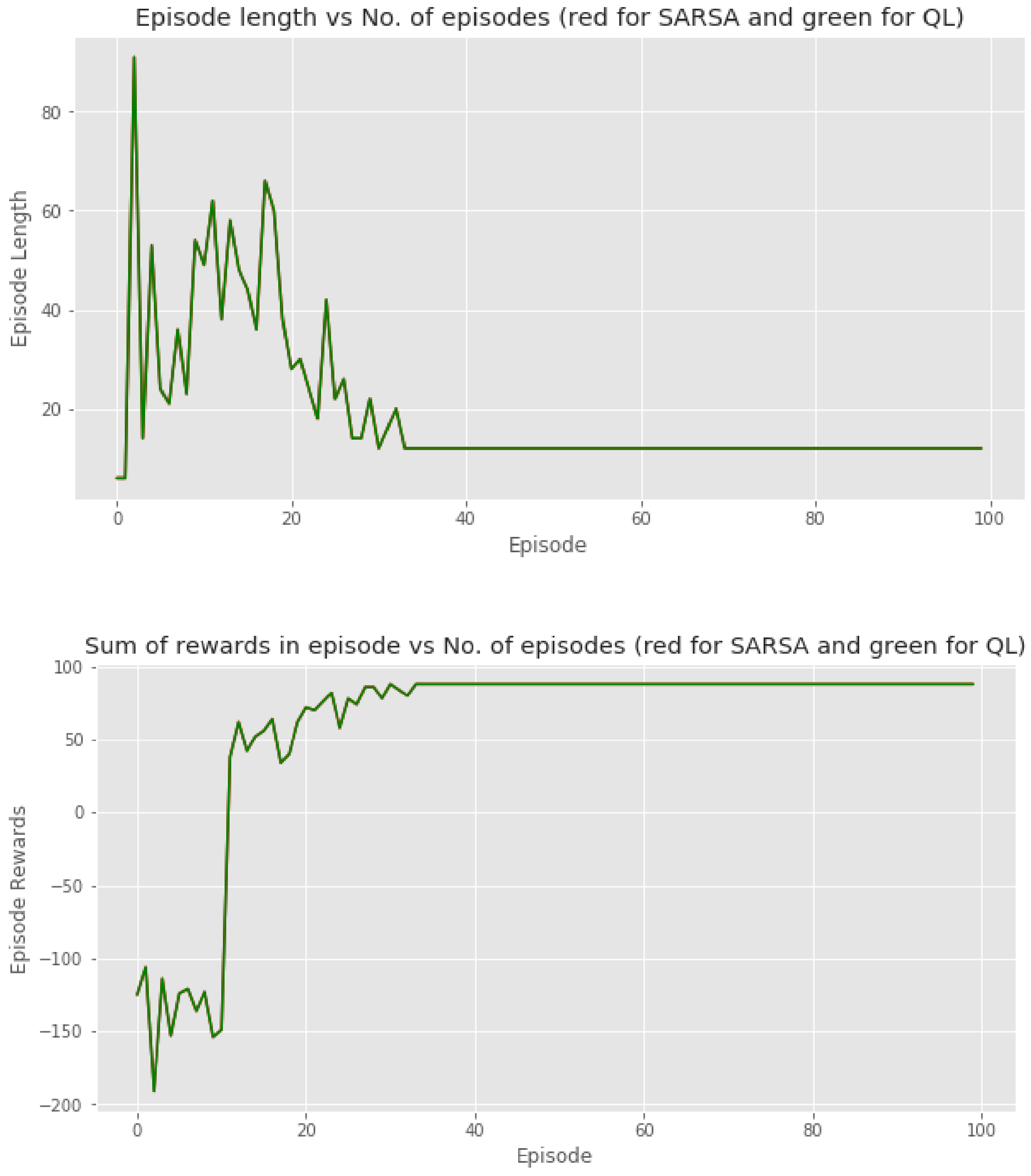


Figure 13: Comparison between Q-learning and SARSA algorithm

5 Observation

- In case-1, Q-Learning is taking lesser average episode length and gaining lesser average rewards in comparison to SARSA in whole learning process. But if we see optimal policy tables then we can observe that SARSA optimal policy is better than Q-learning optimal policy because it's taking path which closer to cliff positions and gaining higher rewards. So SARSA is better than Q-Learning when $\gamma = 0$.

- In case-2, If we see both algorithm optimal policy then we can realize that Q-learning is giving the best optimal path from start state to goal state. Most of the time, Q-learning and SARSA algorithm both are taking episode length less than 20 which faster convergence than case-1. Also, case-2 is collecting reward -100 which is higher rewards comparison to case-1.
- In case-3, Both algorithm are choosing same path from start state to goal state but average episodes length of Q-learning is lesser than SARSA algorithm and average episodes rewards of both algorithms are almost same to each other.
- In case-4, Both algorithm are choosing same path from start state to goal state which is optimal path and the graph of "Episode length vs No. of episodes" and "Sum of rewards in episode vs No. of episodes" of Q-learning and SARSA algorithm are exactly same because due to greedy action selection process, both algorithms are choosing same path which make same episode length and rewards.

6 Questions and Answers

- For each case of parameterization, plot the "Sum of rewards in episode" versus "No. of episodes" for Q-Learning and SARSA algorithms?
  For each case of parameterization, I included plot in section 4 and observation in section 5.
- For each case of parameterization, plot the optimal policy obtained using the Q-Learning and SARSA algorithm?
  For each case of parameterization, I included optimal policy table after graphs of each case in section 4.
- Based on your experimentation, which one has better online performance - Q-Learning or SARSA and why?
  According to my experimentation results, Q-learning is better than SARSA algorithm in case-2 and case-3 but in case-1 where $\gamma = 0$ the SARSA is giving better result in comparison of Q-learning because in Q-learning, algorithm is always choosing first action from action list which making condition more worst. So, Q-Learning has better online performance in cliff walking problem.
- Suppose action selection is greedy. Is Q-learning then exactly the same algorithm as SARSA? Will they make exactly the same action selections and weight updates? Supplement your answer with results from your implementation?
  When action selection is greedy then Both algorithms are work exactly same with same action selection and also weight updates are same in both Q-values table because of weight update equation of Q-learning and SARSA. I included result in case-4 of section 4.