# Reinforcement Learning - CS6700
# PA2 REPORT

*K Pawan Prasad*, **Roll Number: ME16B179**

March 2020

## 1 Question 1: Building the Environment

Origin of grid is set at top left corner:
Therefore Coordinates of Goals A, B and C are: (0,11), (2,9), (6,7) respectively
Start Points are: (5,0), (6,0), (10,0), (11,0)

Wind effect is switched off when implementing algorithm for Goal C

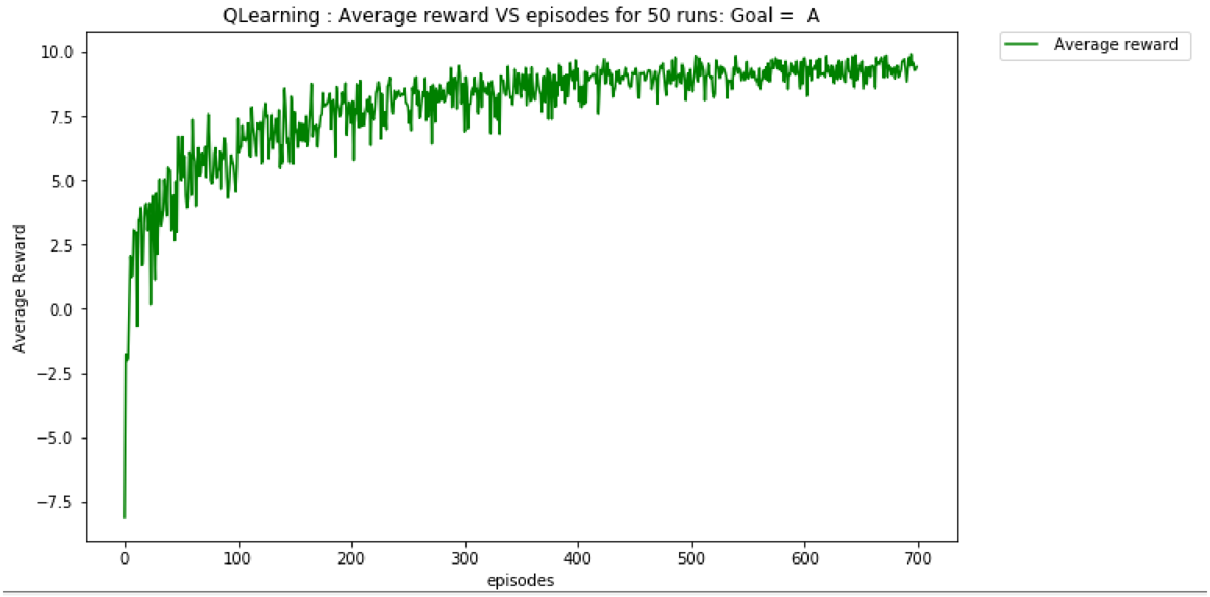
## 2 Question 2: Q-Learning

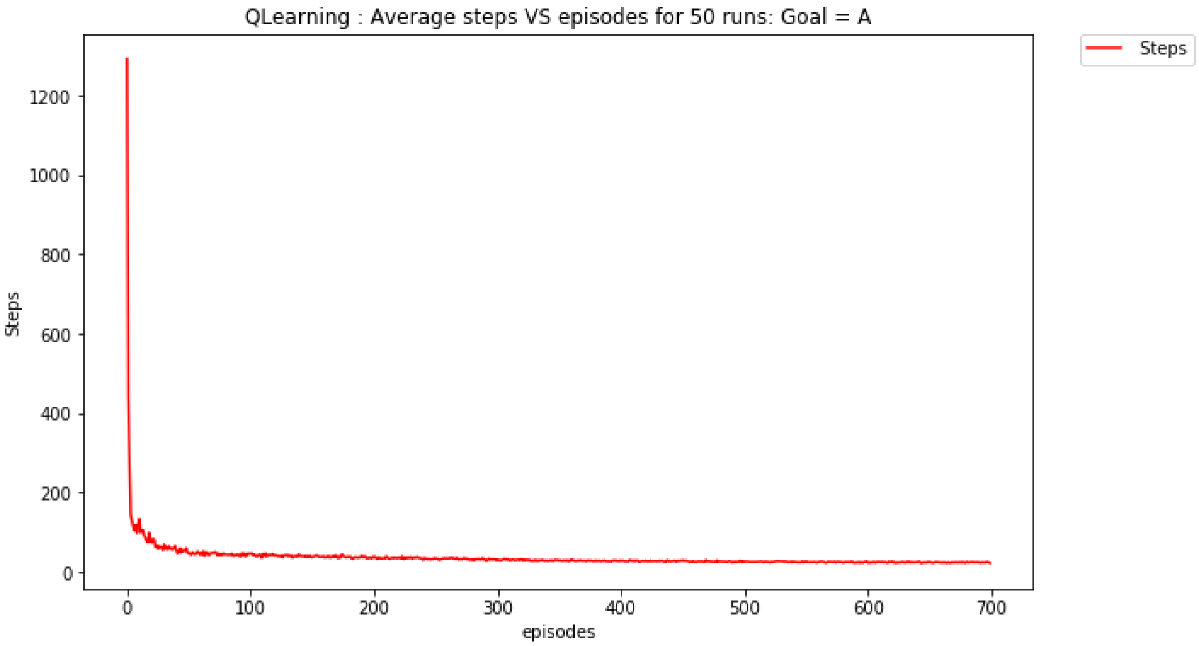
We implement the algorithm over 50 runs and plot average rewards/steps vs episodes for the same. The following are the parameter settings:

$$\alpha = 0.1 \ \epsilon = 0.1 \ \gamma = 0.9$$

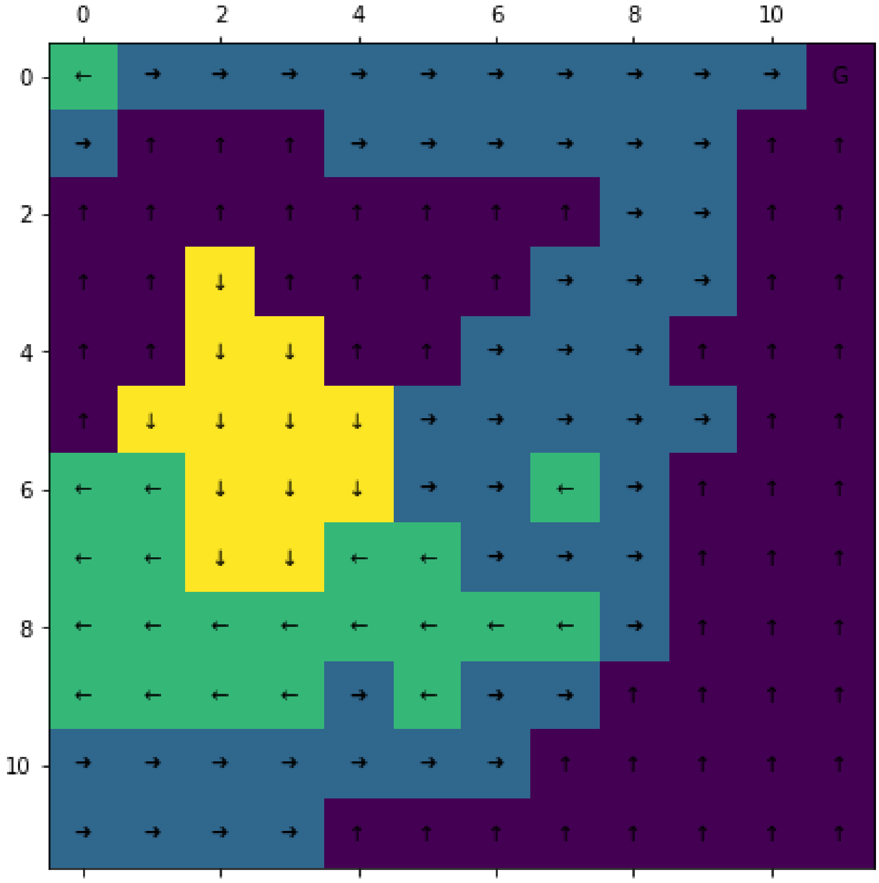
The number of episodes has been fine-tuned and set to **700**. Even larger runs produced better rewards at the cost of program running time.

### 2.1 Plots: Goal A

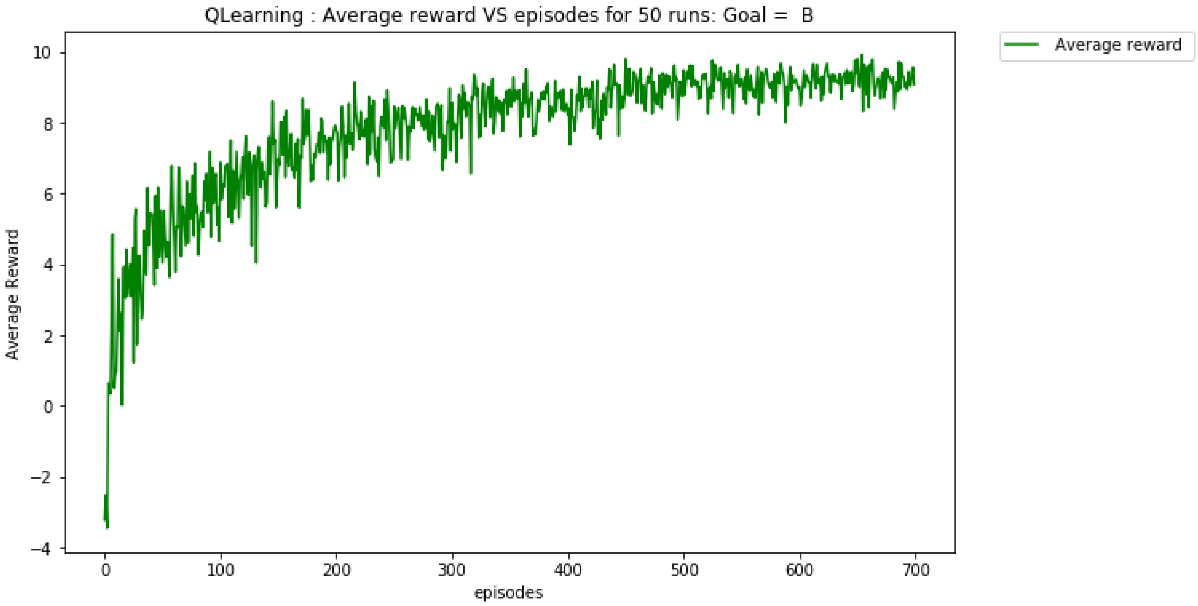


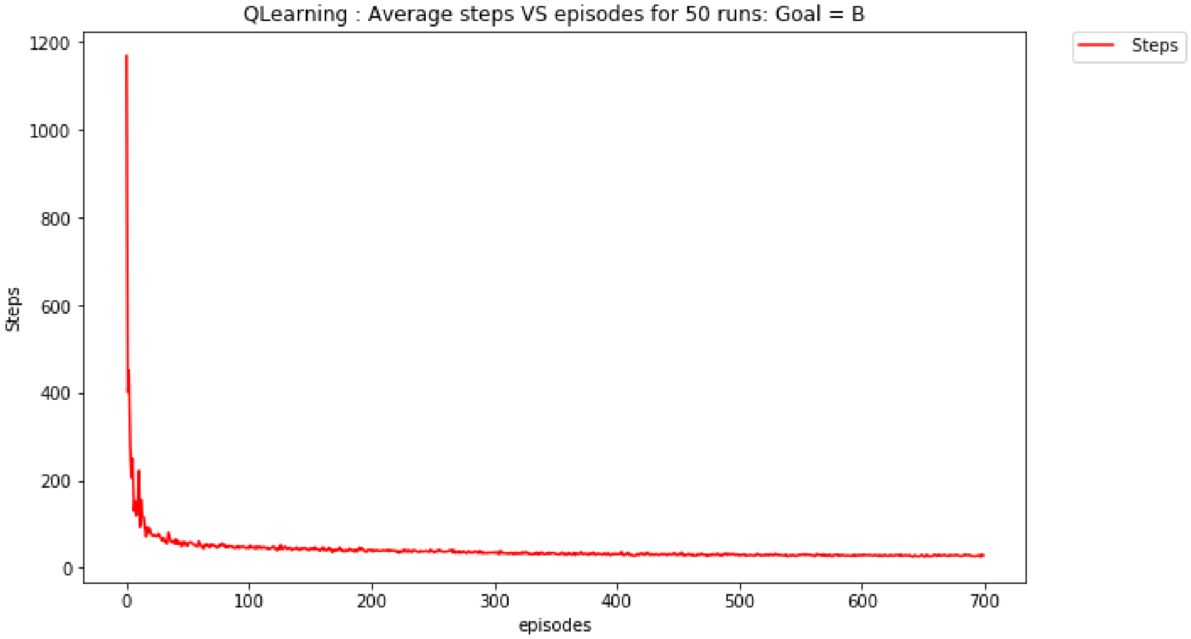


### 2.1.1 Policy Map: Goal A

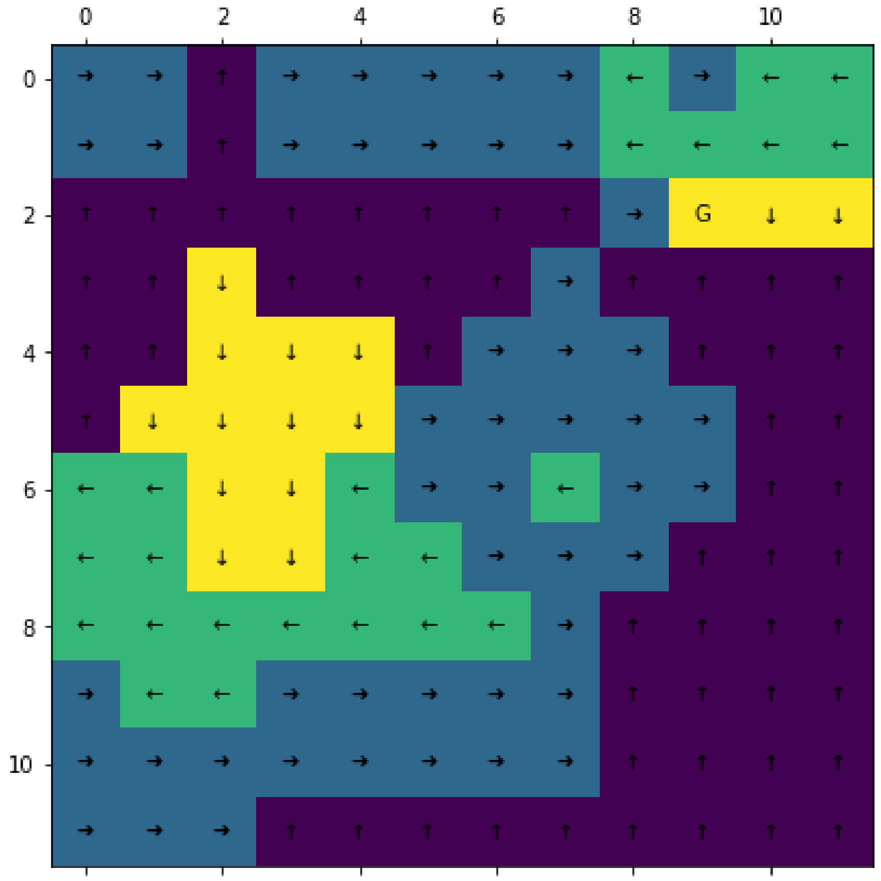


## 2.2 Plots: Goal B

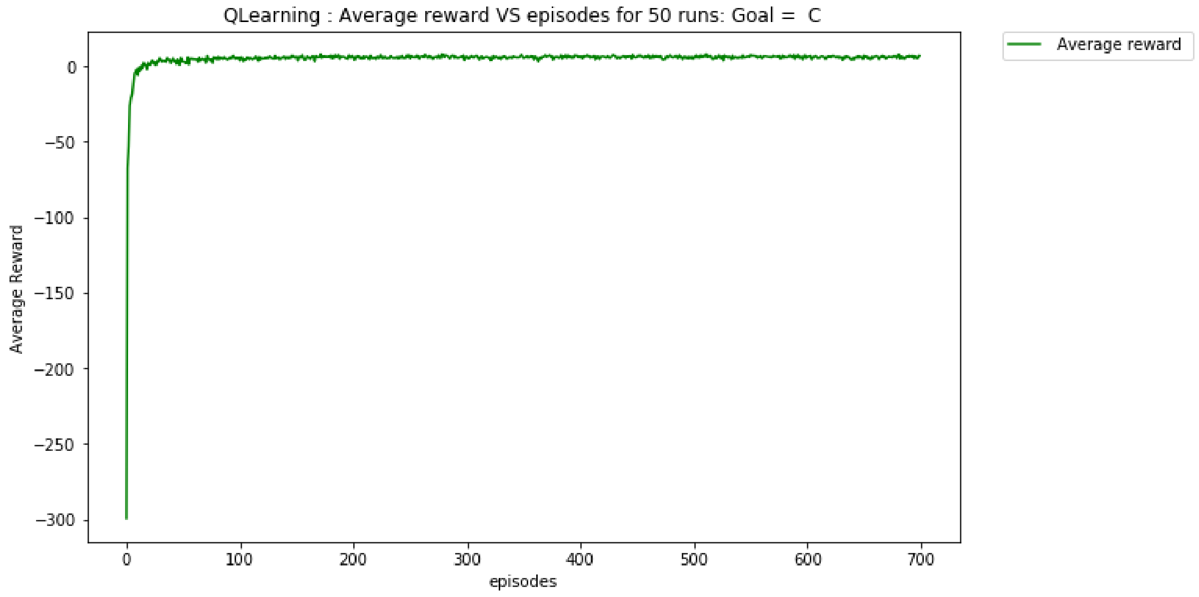


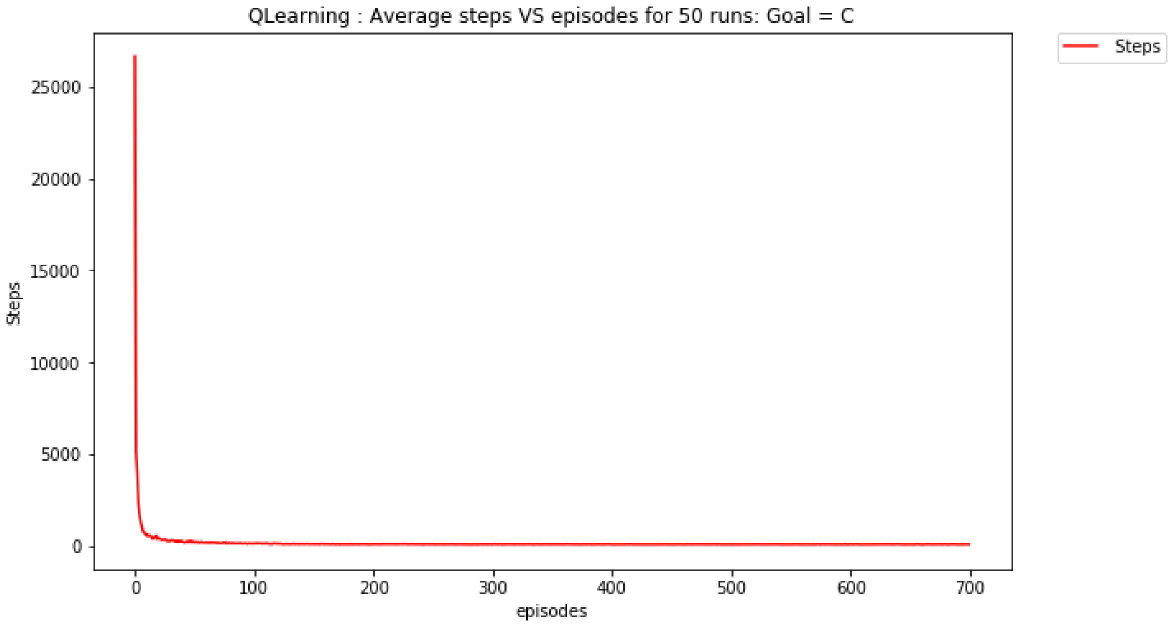


### 2.2.1 Policy Map: Goal B

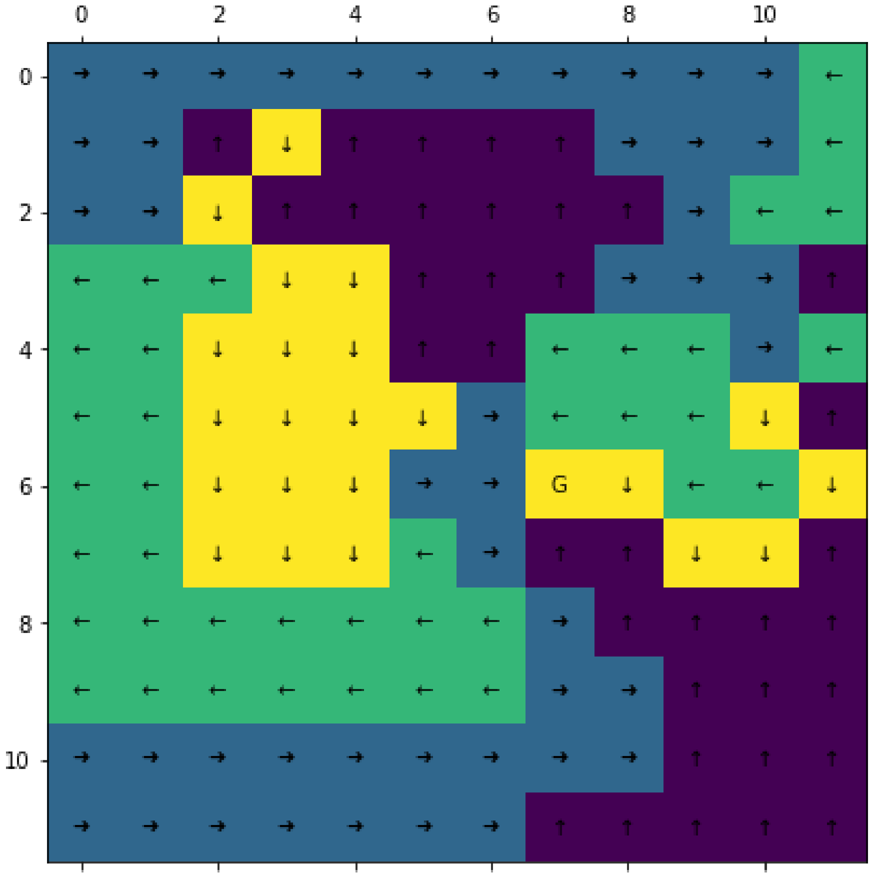


## 2.3 Plots: Goal C





### 2.3.1 Policy Map: Goal C



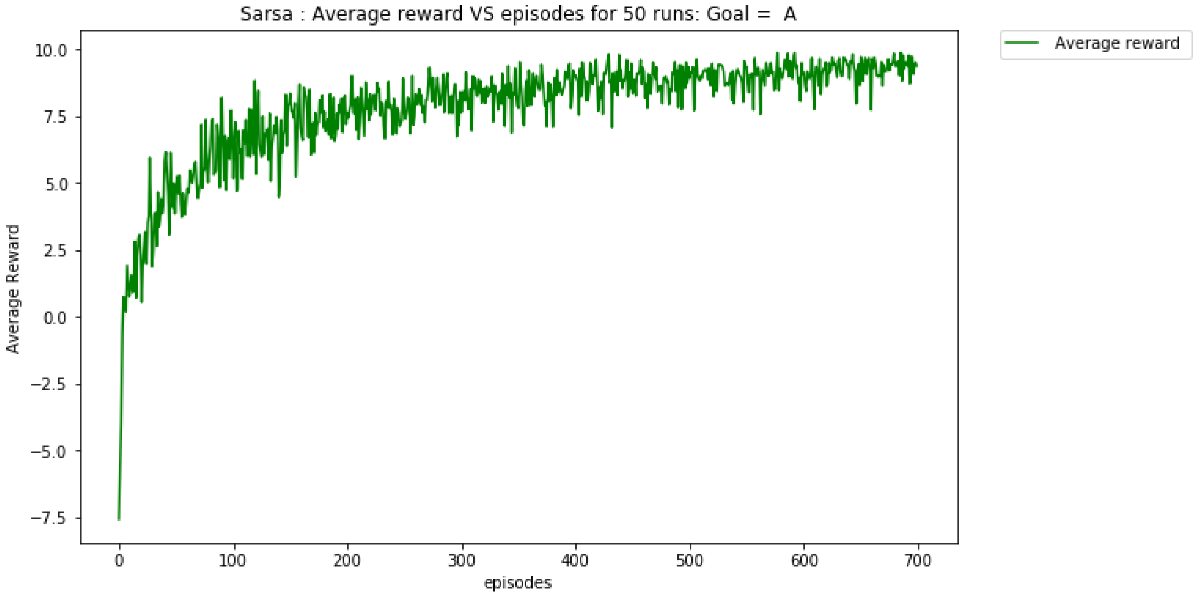
# 3 Question 3: SARSA

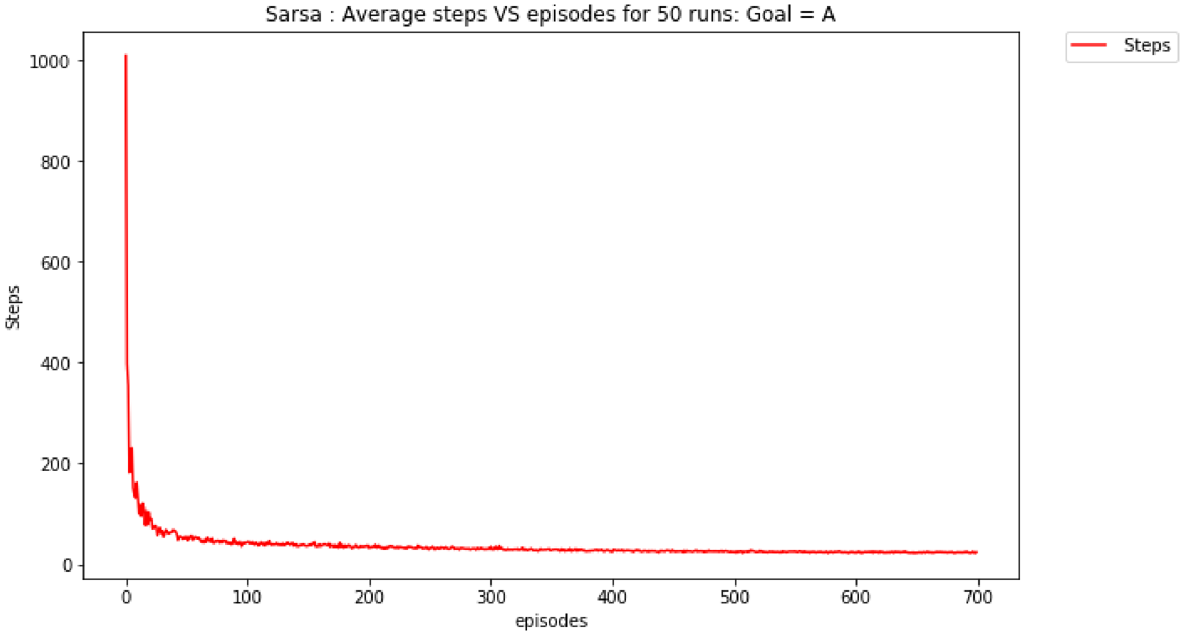
We implement the algorithm over 50 runs and plot average rewards/steps vs episodes for the same. The following are the parameter settings:

$$\alpha = 0.1 \ \epsilon = 0.1 \ \gamma = 0.9$$

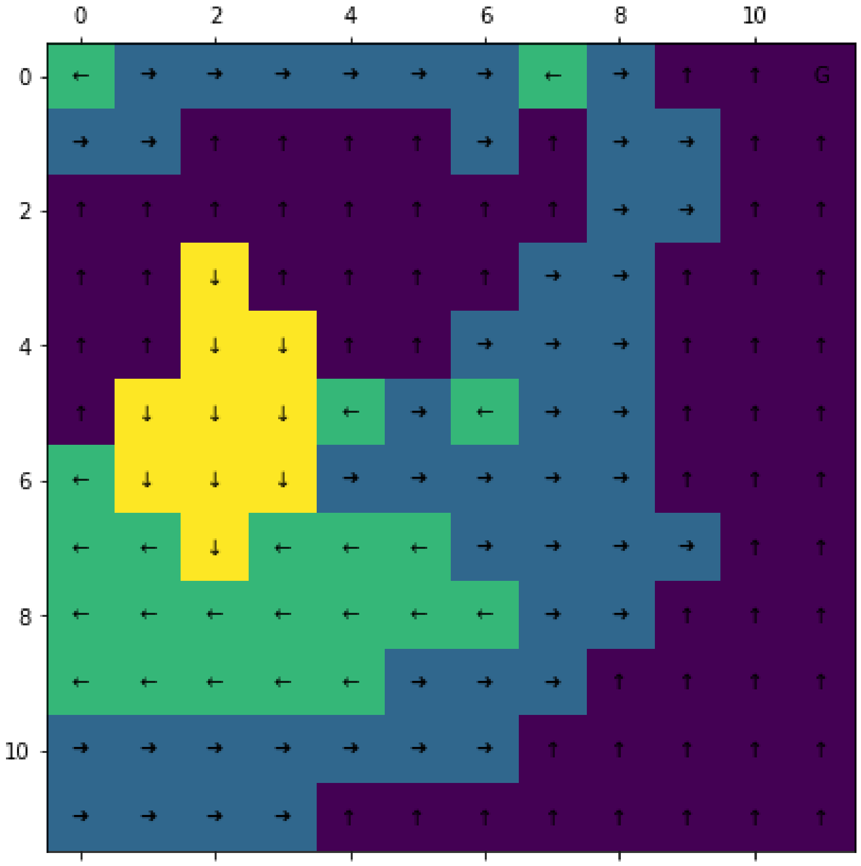
The number of episodes has been fine-tuned and set to **700**. Even larger runs produced better rewards at the cost of program running time.

## 3.1 Plots: Goal A

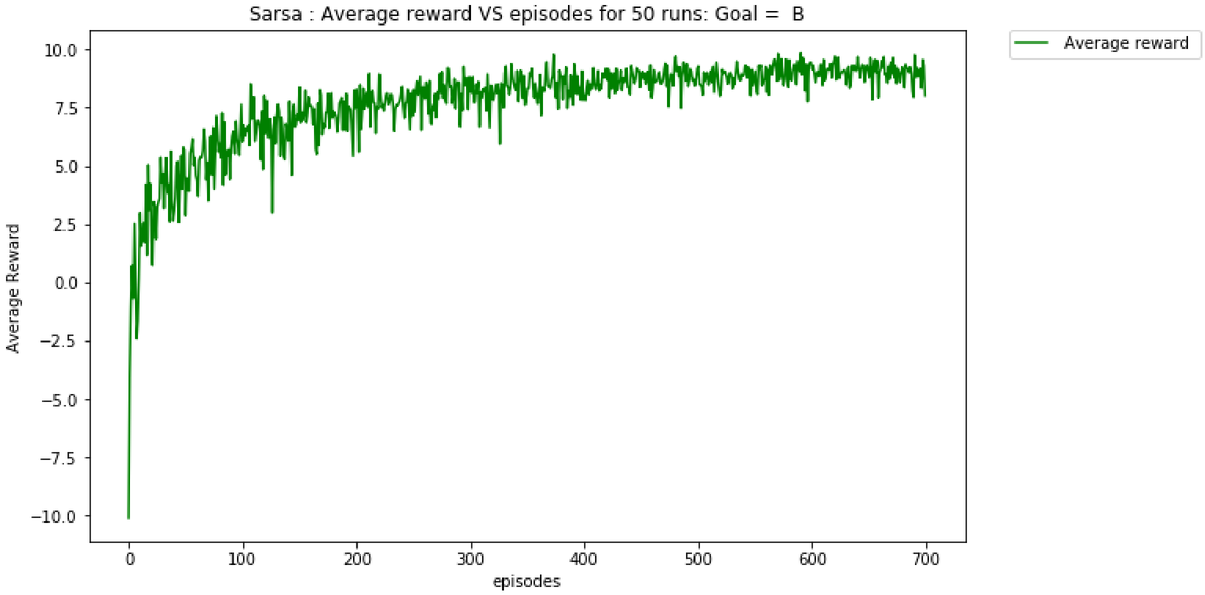


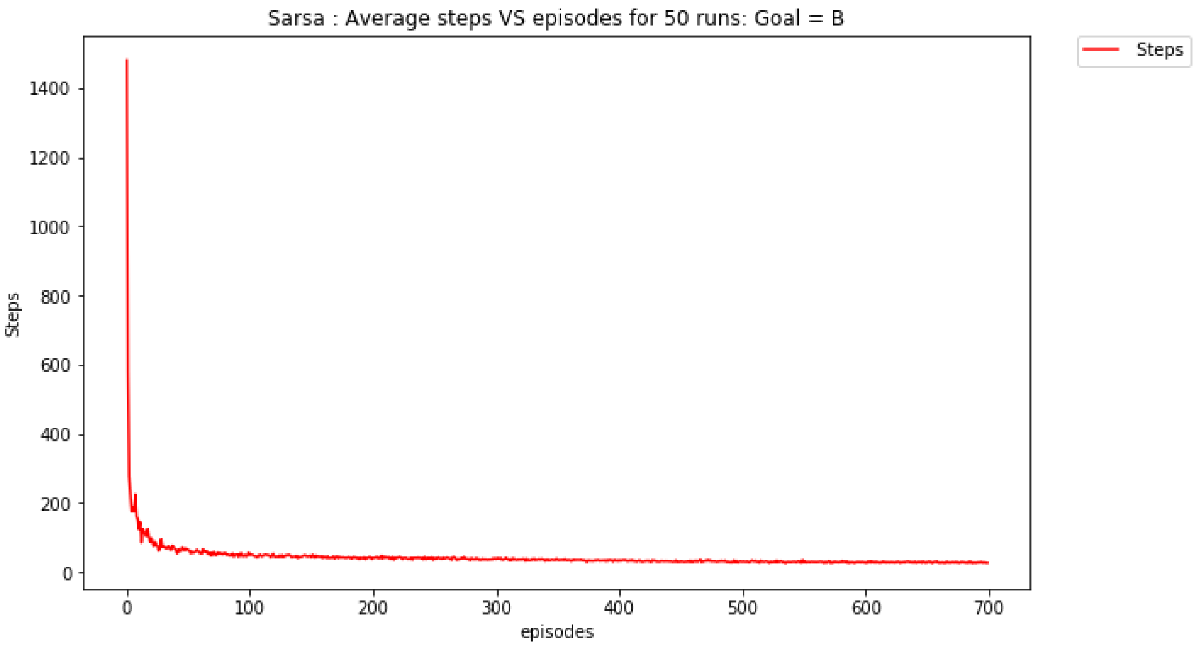


### 3.1.1 Policy Map: Goal A

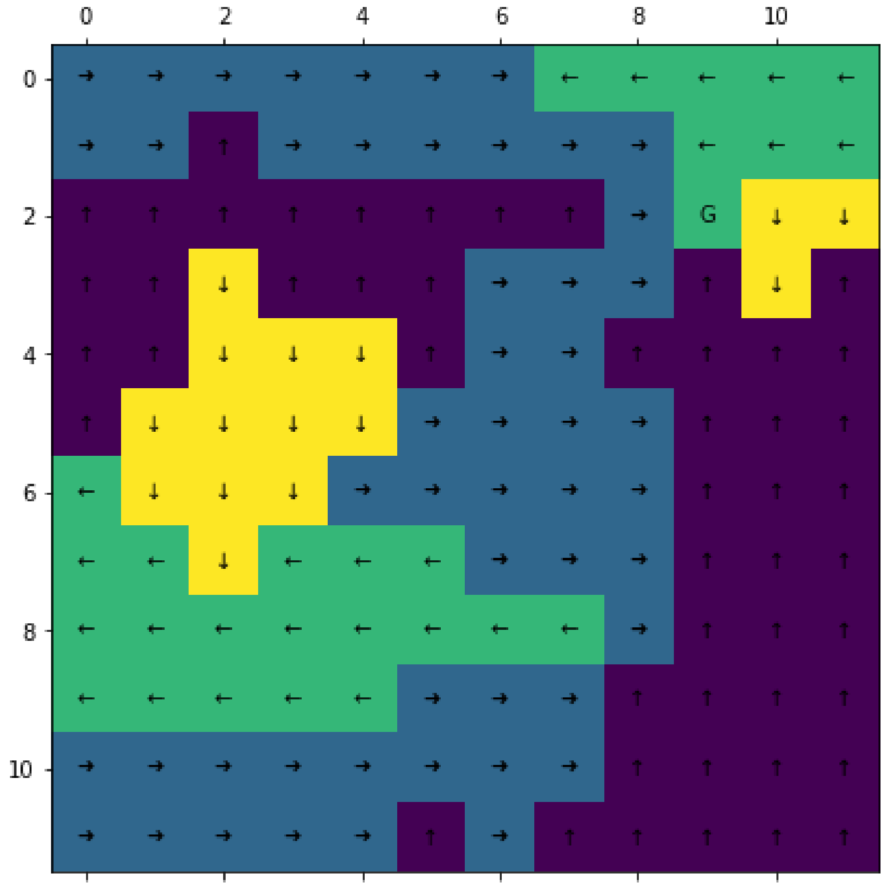


## 3.2 Plots: Goal B

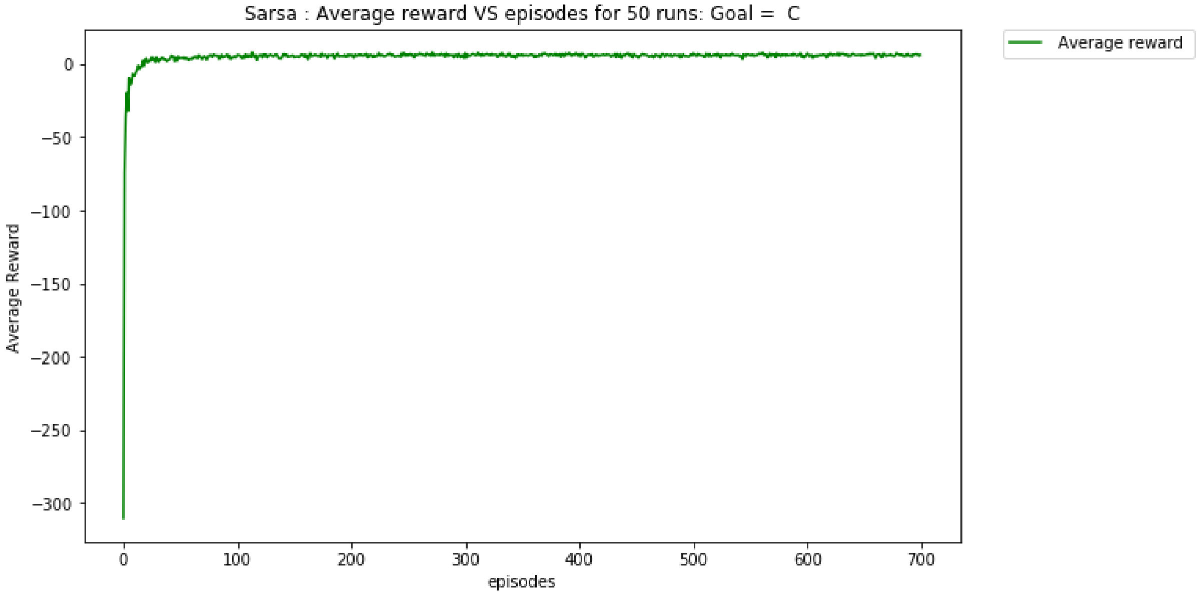


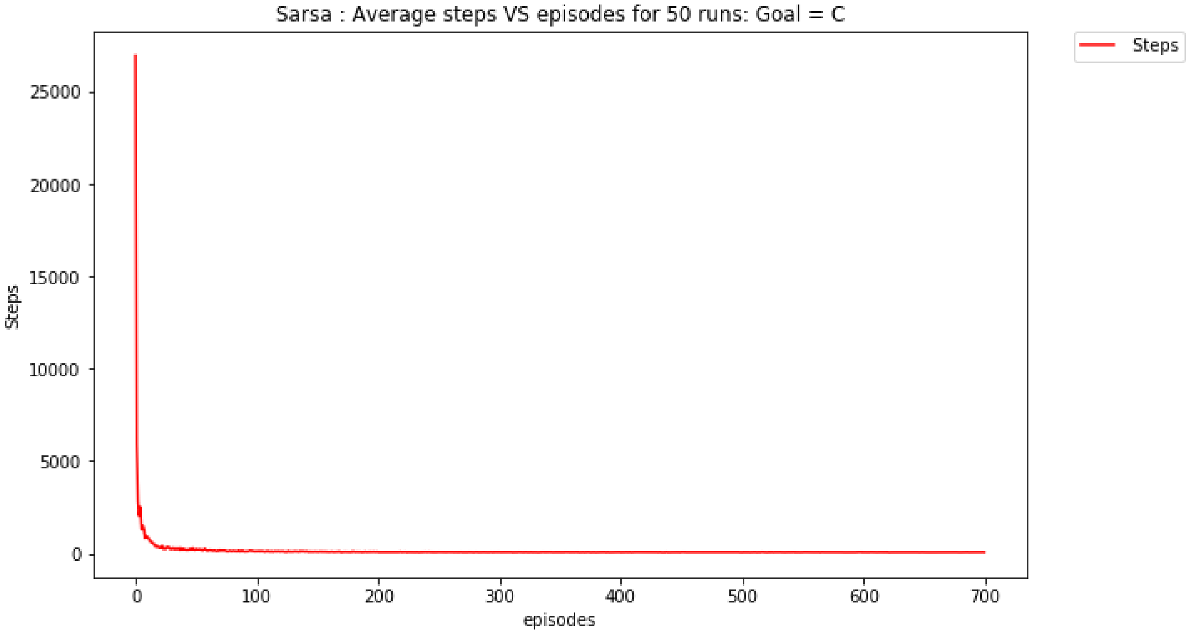


#### 3.2.1 Policy Map: Goal B

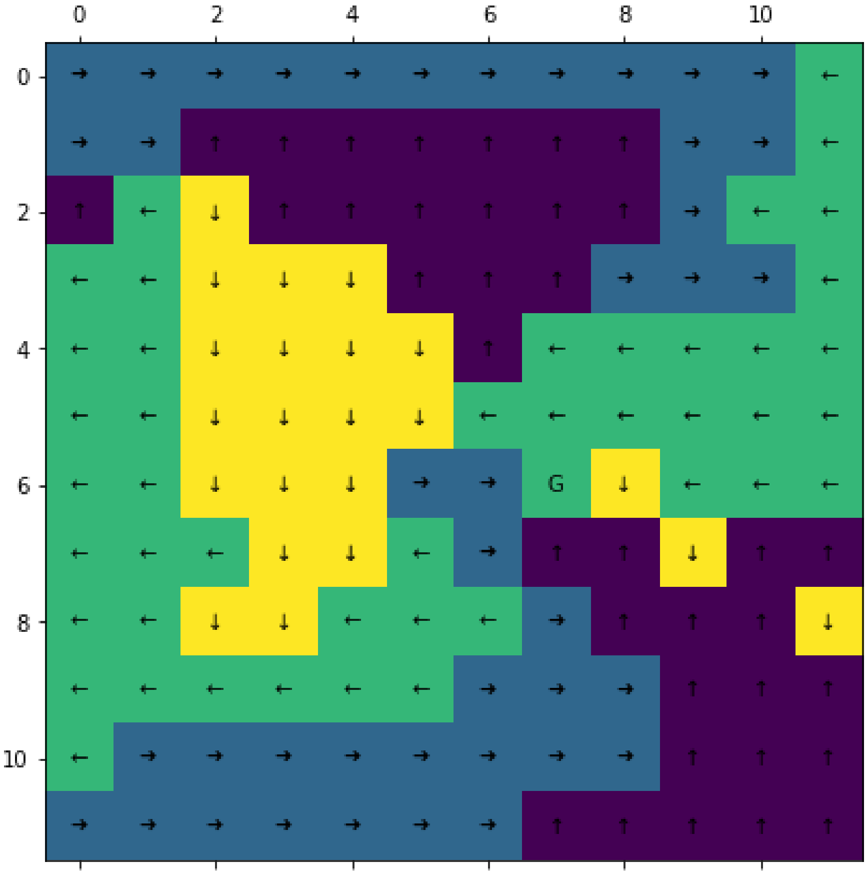


### 3.3 Plots: Goal C





#### 3.3.1 Policy Map: Goal C



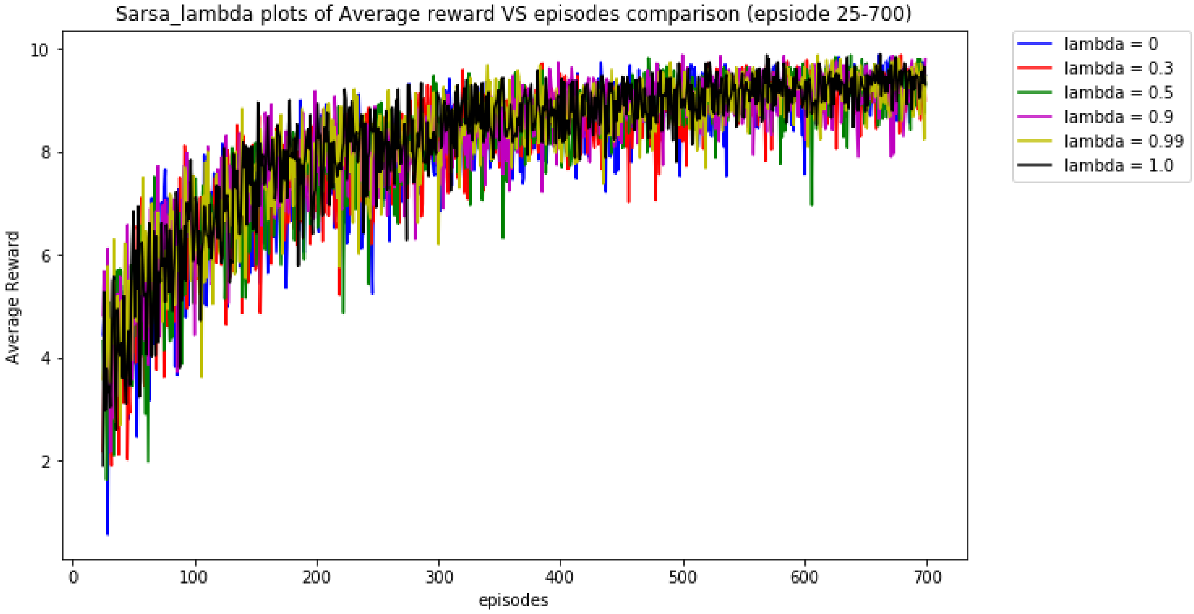
| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| → | → | → | → | → | → | → | → | → | → | → | ← |
| → | → | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | → | → | ← |
| ↑ | ← | ↓ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | → | ← | ← |
| ← | ← | ↓ | ↓ | ↓ | ↑ | ↑ | ↑ | → | → | → | ← |
| ← | ← | ↓ | ↓ | ↓ | ↓ | ↑ | ← | ← | ← | ← | ← |
| ← | ← | ↓ | ↓ | ↓ | ↓ | ← | ← | ← | ← | ← | ← |
| ← | ← | ↓ | ↓ | ↓ | → | → | G | ↓ | ← | ← | ← |
| ← | ← | ← | ↓ | ↓ | ← | → | ↑ | ↑ | ↓ | ↑ | ↑ |
| ← | ← | ↓ | ↓ | ← | ← | ← | → | ↑ | ↑ | ↑ | ↓ |
| ← | ← | ← | ← | ← | ← | → | → | → | ↑ | ↑ | ↑ |
| ← | → | → | → | → | → | → | → | → | ↑ | ↑ | ↑ |
| → | → | → | → | → | → | → | ↑ | ↑ | ↑ | ↑ | ↑ |

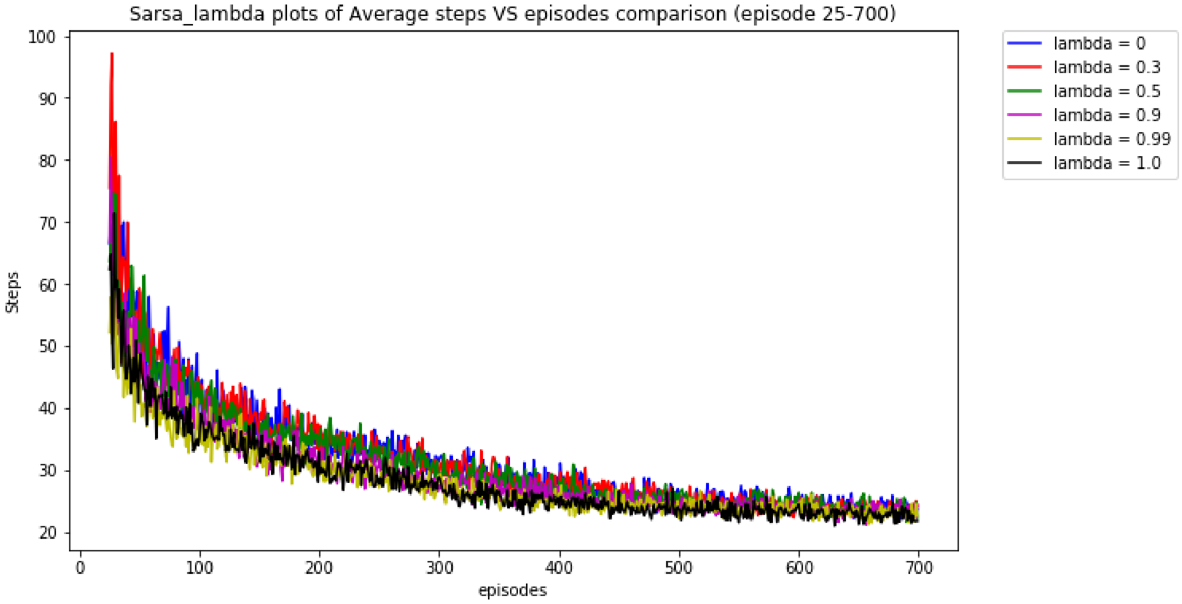
# 4 Question 4: SARSA-$\lambda$

The comparison plots after 25 learning trials are shown for the following values of Lambda:

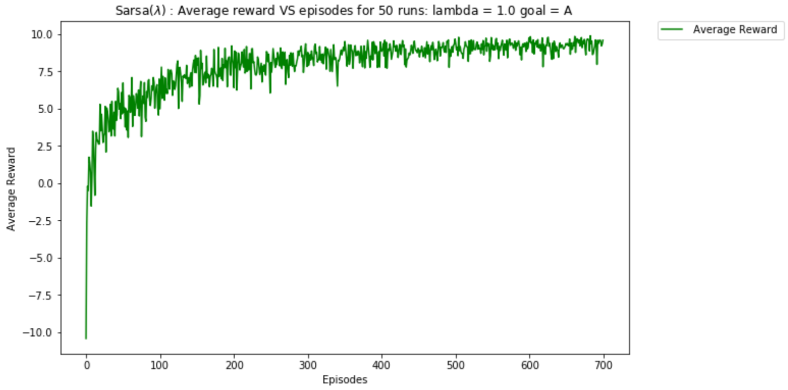
$$\lambda = \{\ 0,\ 0.3,\ 0.5,\ 0.9,\ 0.99,\ 1.0\ \}$$

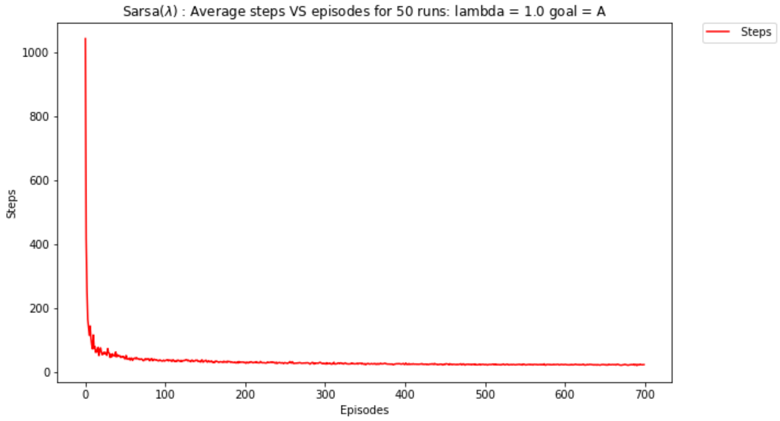
## 4.1 Comparison Plots: Goal A

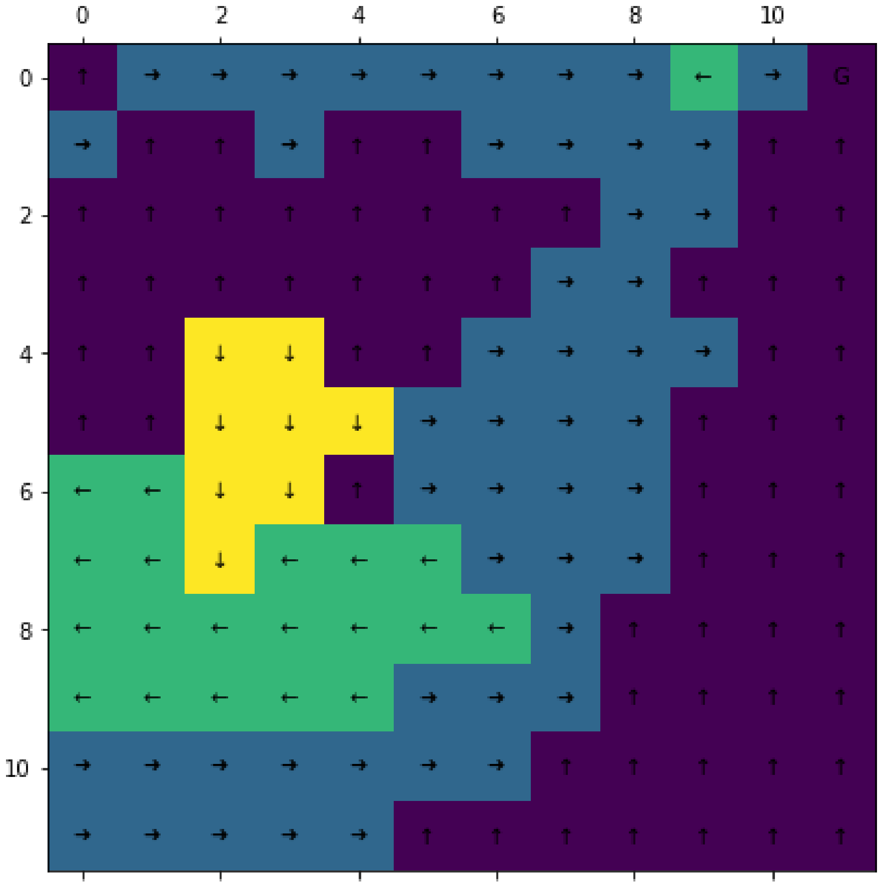




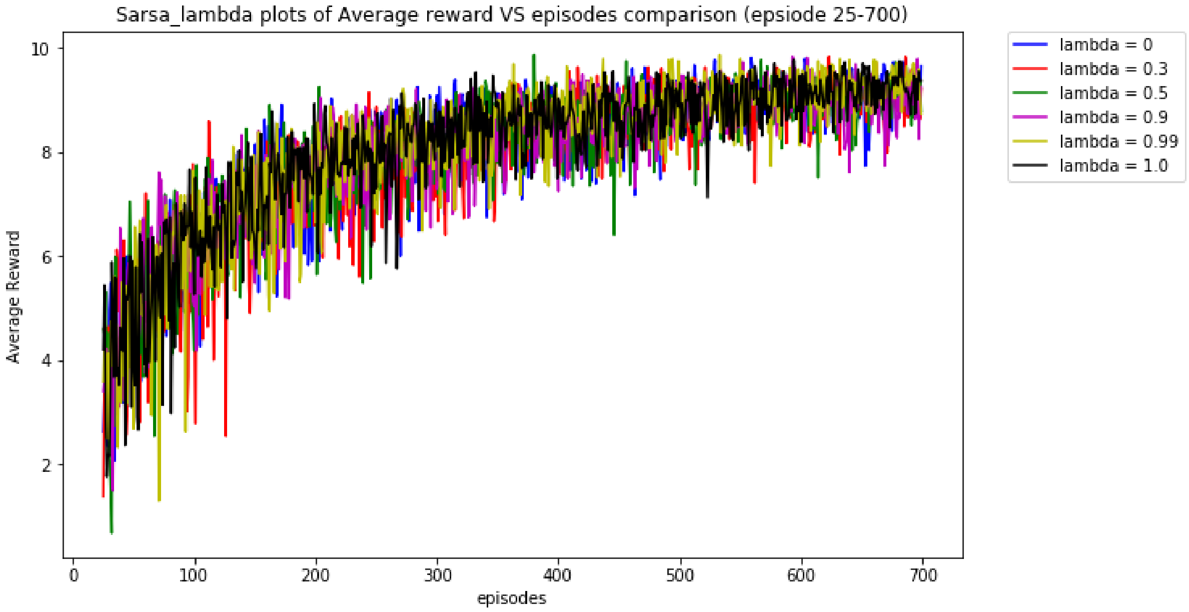
We observe from the above 2 plots that the best setting is $\lambda = 1$. We provide the individual plots and policy map for the same:

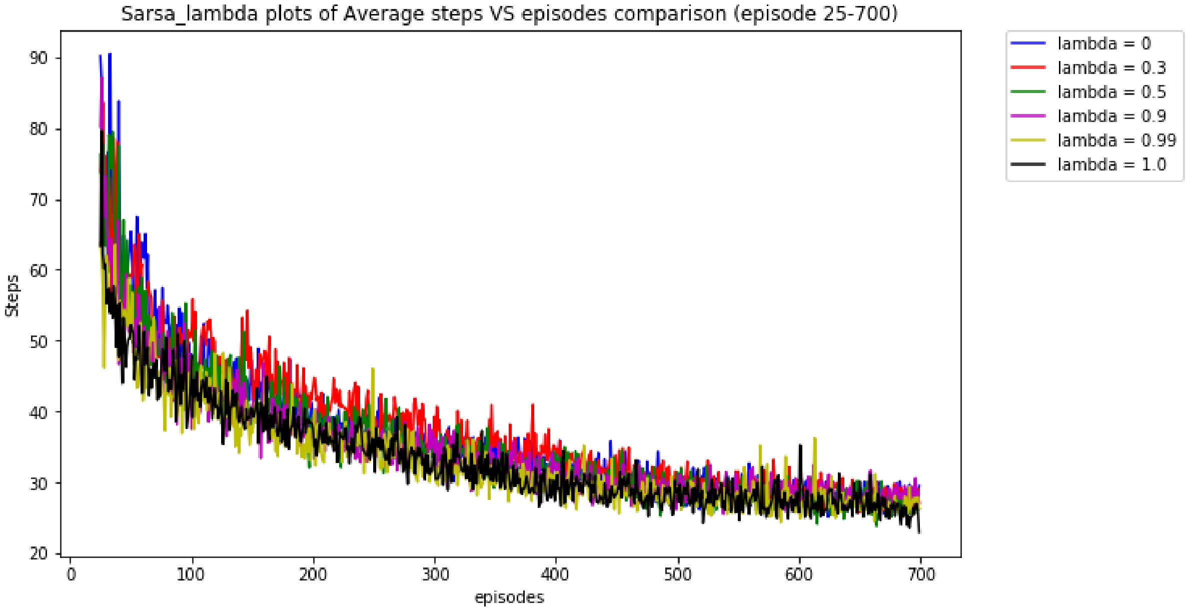

Sarsa(λ) : Average reward VS episodes for 50 runs: lambda = 1.0 goal = A
Average Reward
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
0
100
200
300
400
500
600
700
Episodes
Average Reward


Sarsa(λ) : Average steps VS episodes for 50 runs: lambda = 1.0 goal = A
Steps
1000
800
600
400
200
0
0
100
200
300
400
500
600
700
Episodes
Steps

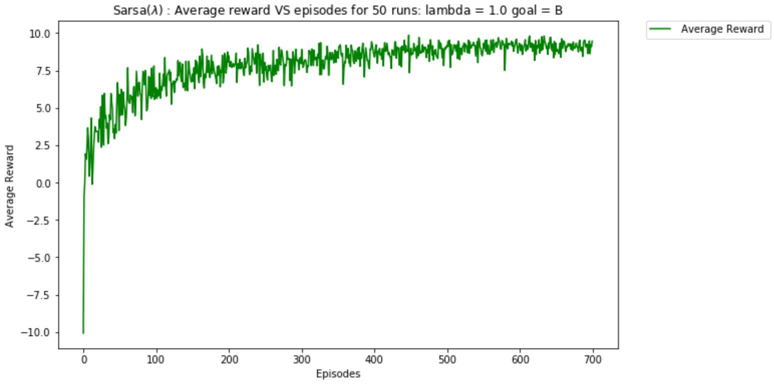

0
2
4
6
8
10
0
2
4
6
8
10
G

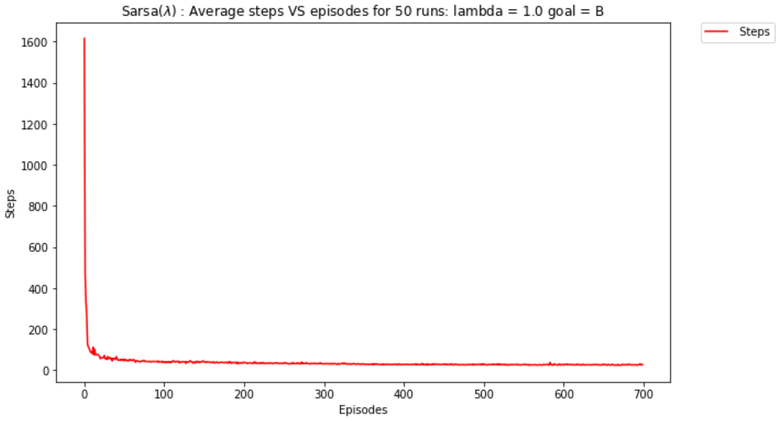
## 4.2 Comparison Plots: Goal B

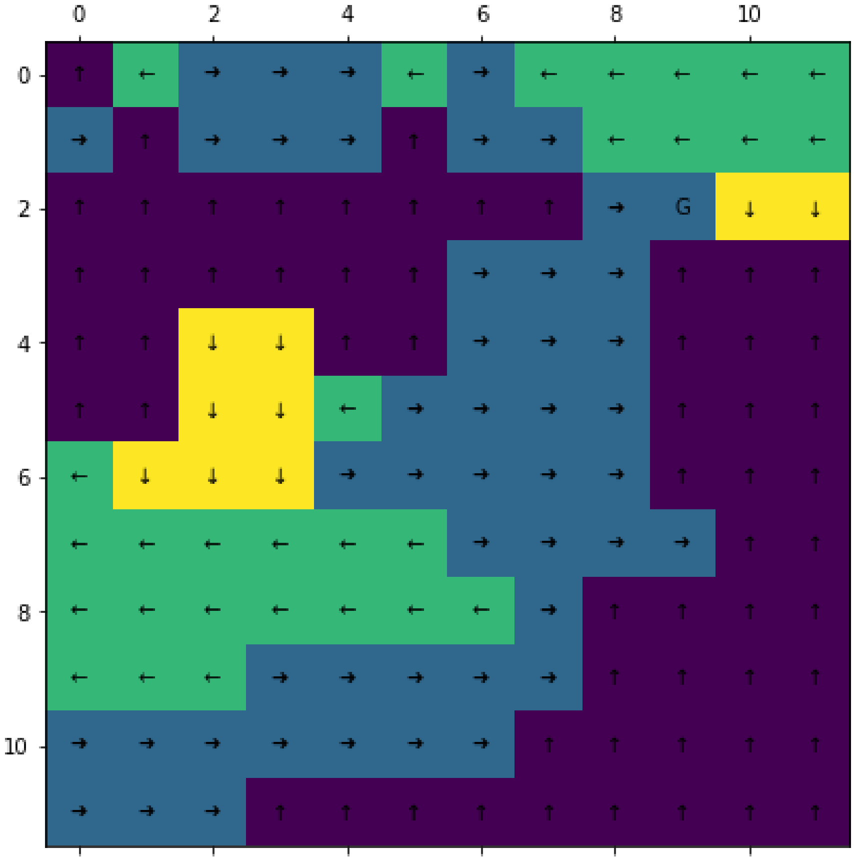




We observe from the above 2 plots that the best setting is $\lambda = 1$. We provide the individual plots and policy map for the same:

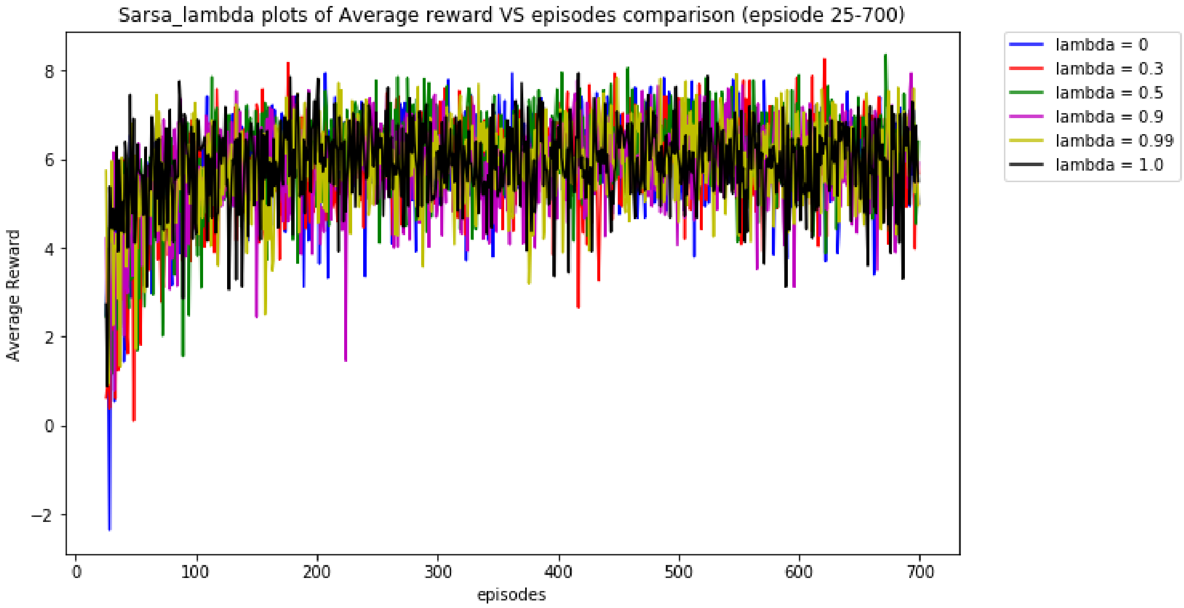

Sarsa($\lambda$) : Average reward VS episodes for 50 runs: lambda = 1.0 goal = B
Average Reward
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
0
100
200
300
400
500
600
700
Episodes
Average Reward

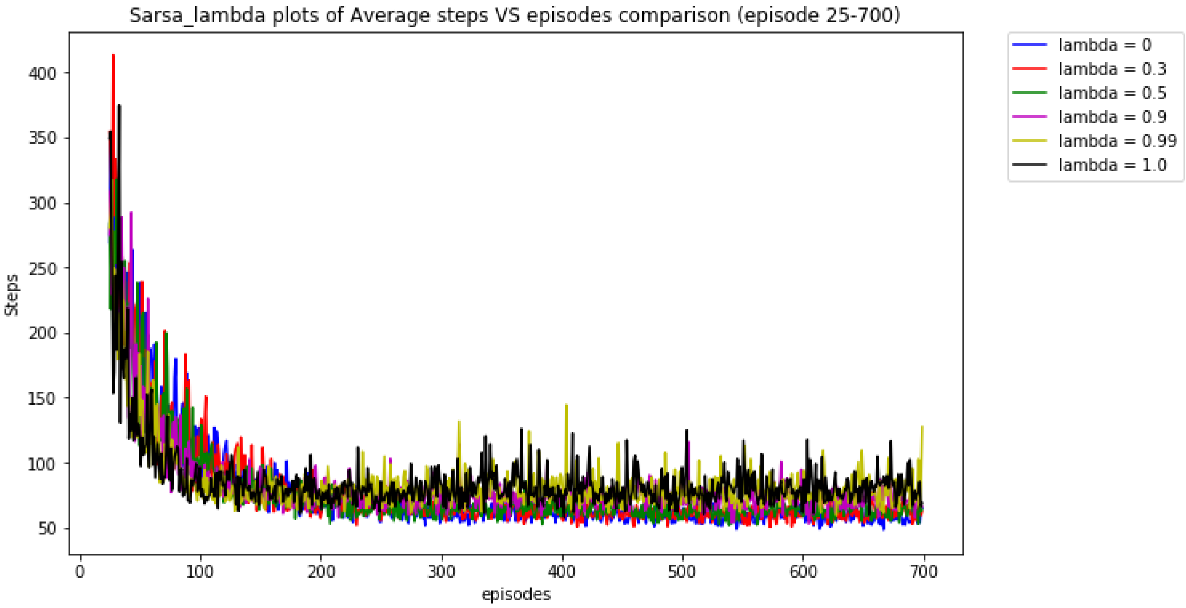

Sarsa($\lambda$) : Average steps VS episodes for 50 runs: lambda = 1.0 goal = B
Steps
1600
1400
1200
1000
800
600
400
200
0
0
100
200
300
400
500
600
700
Episodes
Steps


0
2
4
6
8
10
0
2
4
6
8
10

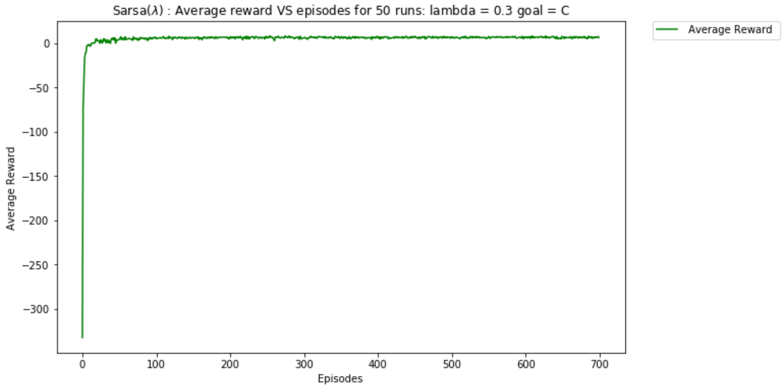
| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ↑ | ← | → | → | → | ← | → | ← | ← | ← | ← | ← |
| → | ↑ | → | → | → | ↑ | → | → | ← | ← | ← | ← |
| ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | → | G | ↓ | ↓ |
| ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | → | → | → | ↑ | ↑ | ↑ |
| ↑ | ↑ | ↓ | ↓ | ↑ | ↑ | → | → | → | ↑ | ↑ | ↑ |
| ↑ | ↑ | ↓ | ↓ | ← | → | → | → | → | ↑ | ↑ | ↑ |
| ← | ↓ | ↓ | ↓ | → | → | → | → | → | ↑ | ↑ | ↑ |
| ← | ← | ← | ← | ← | ← | → | → | → | → | ↑ | ↑ |
| ← | ← | ← | ← | ← | ← | ← | → | ↑ | ↑ | ↑ | ↑ |
| ← | ← | ← | → | → | → | → | → | ↑ | ↑ | ↑ | ↑ |
| → | → | → | → | → | → | → | ↑ | ↑ | ↑ | ↑ | ↑ |
| → | → | → | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ |

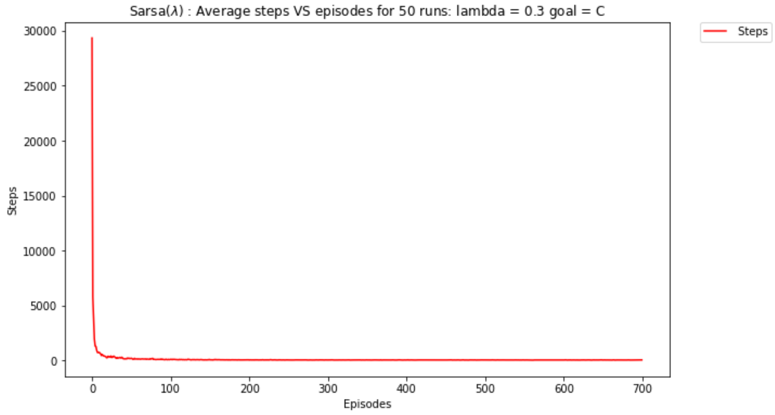
## 4.3 Comparison Plots: Goal C

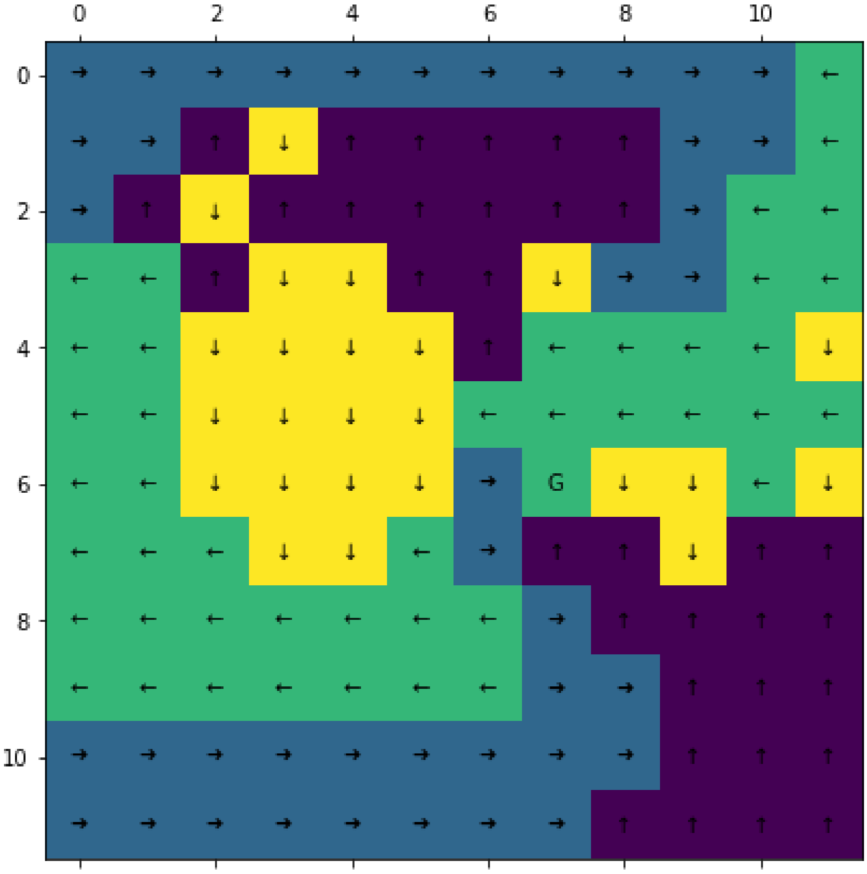




We observe from the above 2 plots that the best setting is $\lambda = 0.3$. We provide the individual plots and policy map for the same:


Sarsa(λ) : Average reward VS episodes for 50 runs: lambda = 0.3 goal = C
Average Reward
0
−50
−100
−150
−200
−250
−300
0
100
200
300
400
500
600
700
Episodes
Average Reward


Sarsa(λ) : Average steps VS episodes for 50 runs: lambda = 0.3 goal = C
Steps
30000
25000
20000
15000
10000
5000
0
0
100
200
300
400
500
600
700
Episodes
Steps


0
2
4
6
8
10
0
2
4
6
8
10

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| → | → | → | → | → | → | → | → | → | → | → | ← |
| → | → | ↑ | ↓ | ↑ | ↑ | ↑ | ↑ | ↑ | → | → | ← |
| → | ↑ | ↓ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | → | ← | ← |
| ← | ← | ↑ | ↓ | ↓ | ↑ | ↑ | ↓ | → | → | ← | ← |
| ← | ← | ↓ | ↓ | ↓ | ↓ | ↑ | ← | ← | ← | ← | ↓ |
| ← | ← | ↓ | ↓ | ↓ | ↓ | ← | ← | ← | ← | ← | ← |
| ← | ← | ↓ | ↓ | ↓ | ↓ | → | G | ↓ | ↓ | ← | ↓ |
| ← | ← | ← | ↓ | ↓ | ← | → | ↑ | ↑ | ↓ | ↑ | ↑ |
| ← | ← | ← | ← | ← | ← | ← | → | ↑ | ↑ | ↑ | ↑ |
| ← | ← | ← | ← | ← | ← | ← | → | → | ↑ | ↑ | ↑ |
| → | → | → | → | → | → | → | → | → | ↑ | ↑ | ↑ |
| → | → | → | → | → | → | → | → | ↑ | ↑ | ↑ | ↑ |

# 5 PART 2: Policy Gradients

## 5.1 The environments

The step function has been completed to update the current state and also obtain the corresponding reward in Chakra and vishamC worlds. (Refer the respective .py files)
A reward of +5 has been allotted if the new state is within some given tolerance around origin, else the reward is negative of norm.

## 5.2 The Roll- out function

In the chakra_get_action method, the action steps in $x$ and $y$ have been clipped, that is, if the step size is greater than 0.025 then we clip it to 0.025.
Also added is the include_bias method