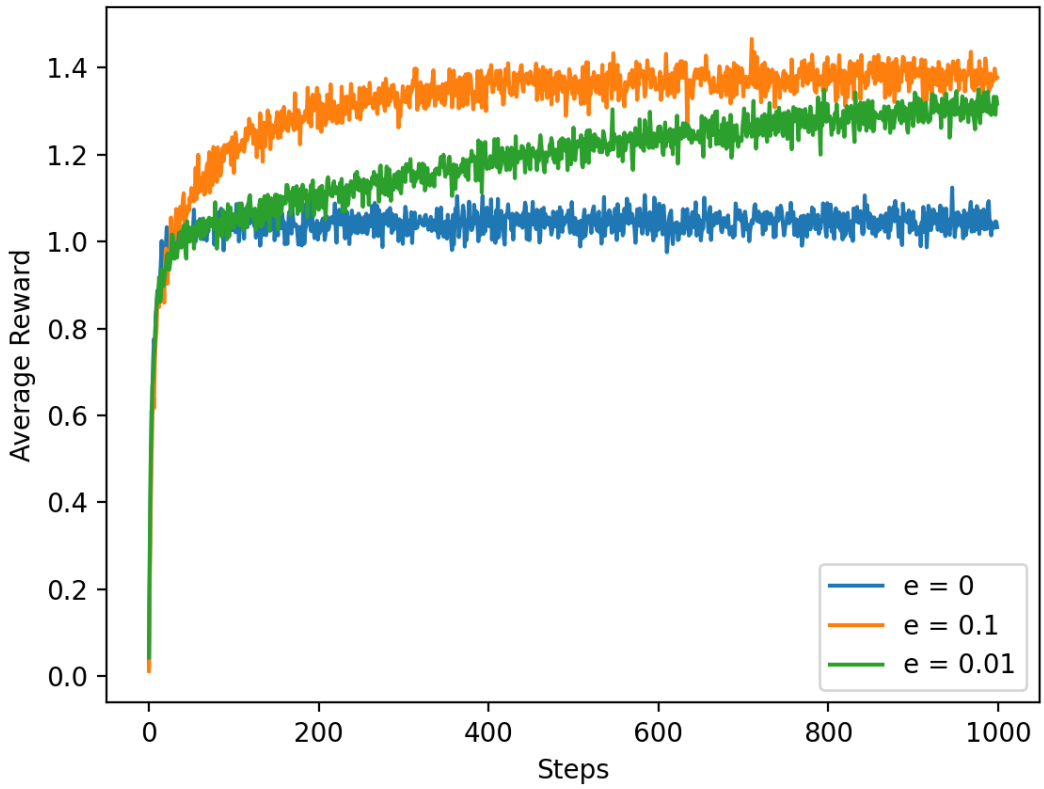
RL – Reinforcement Learning
CSE564
Homework 1

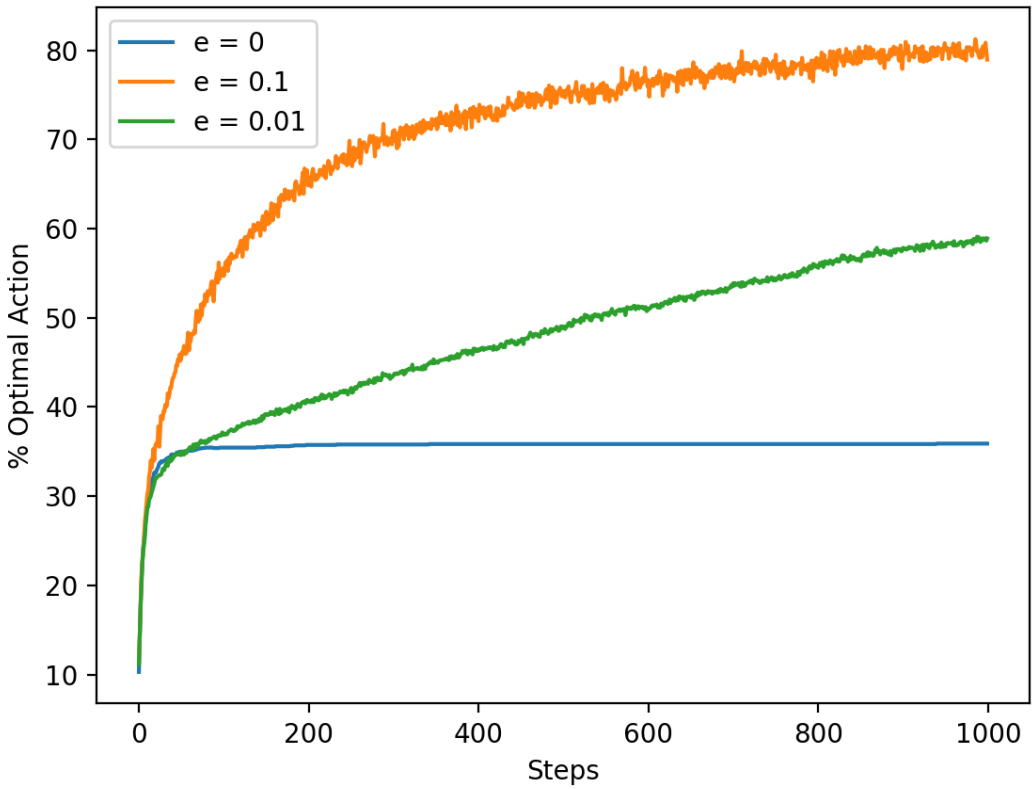
Name: Ansh Arora
Roll No: 2019022

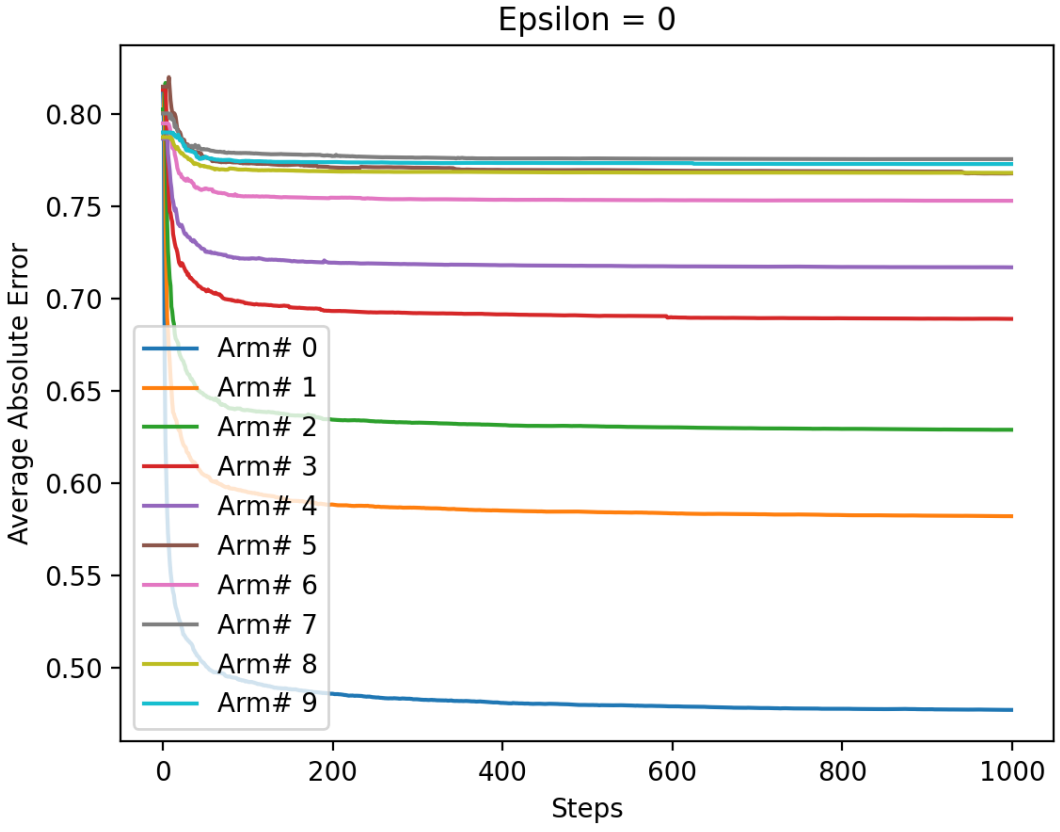
Graphs pasted at beginning, theory Qs at end.

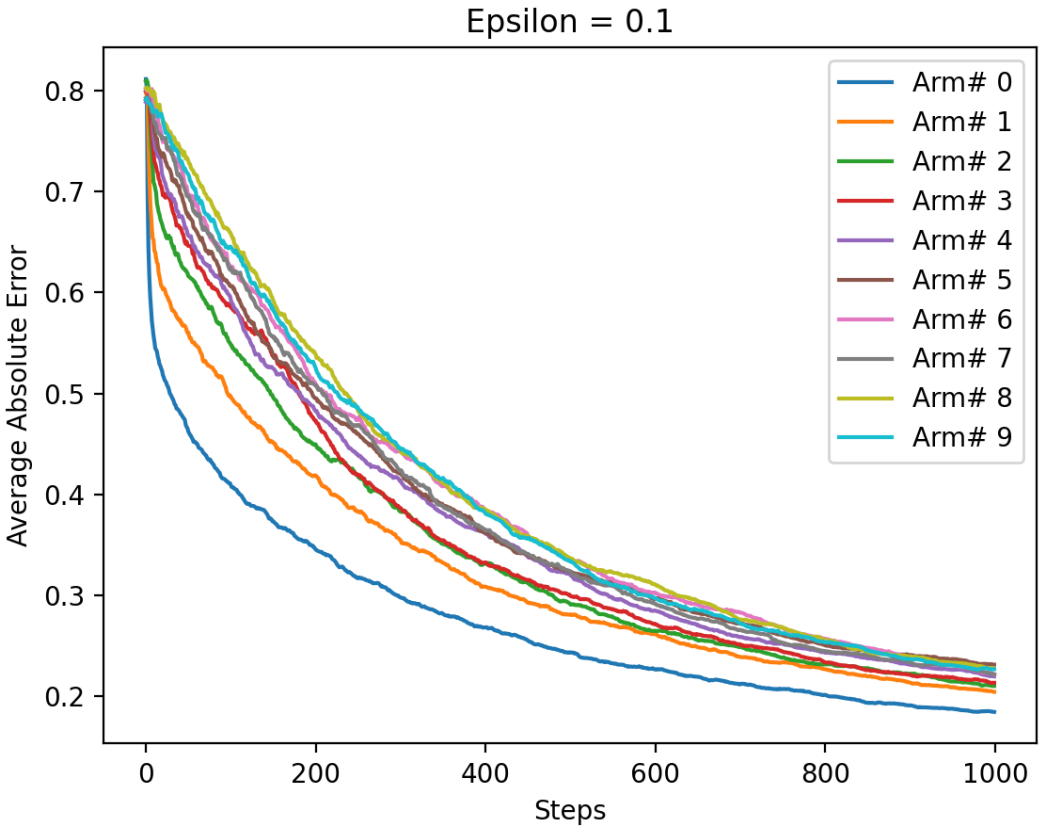
Question 1

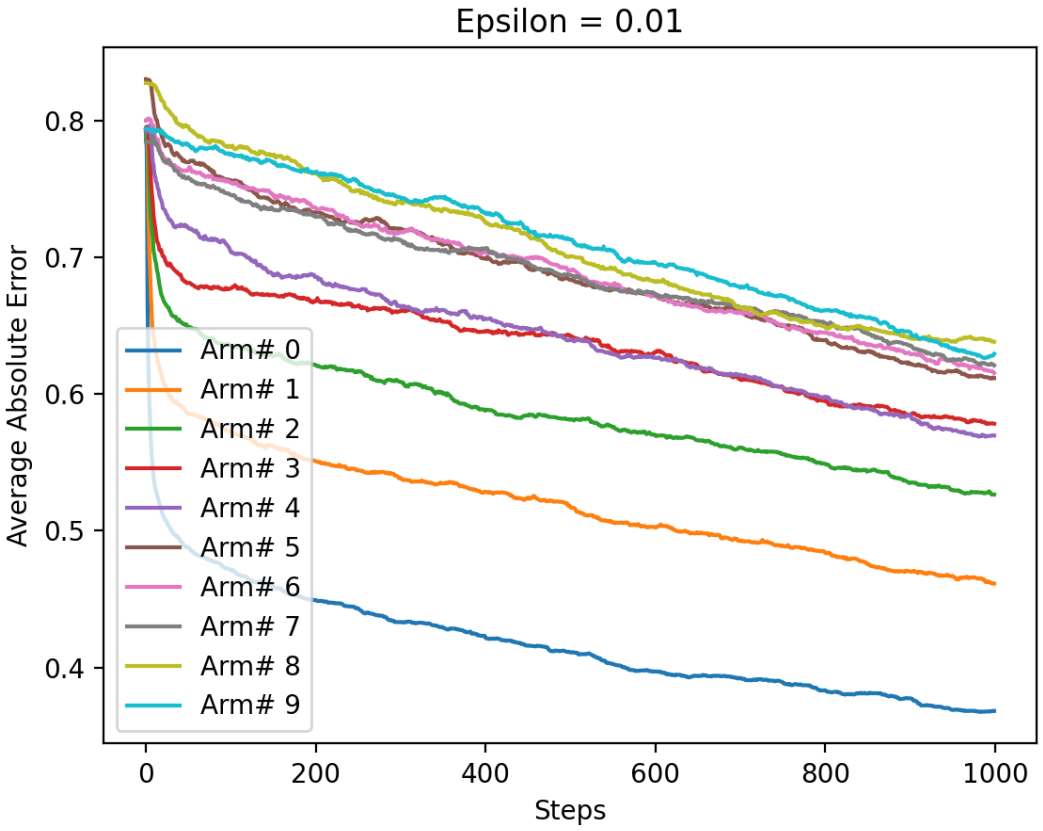
For epsilon = 0, 0.1, 0.01
https://github.com/arora-ansh/RL-M2021_Ansh_2019022/blob/main/HW1/Q1/Q1.py




e = 0
e = 0.1
e = 0.01
% Optimal Action
Steps
10
20
30
40
50
60
70
80
0
200
400
600
800
1000

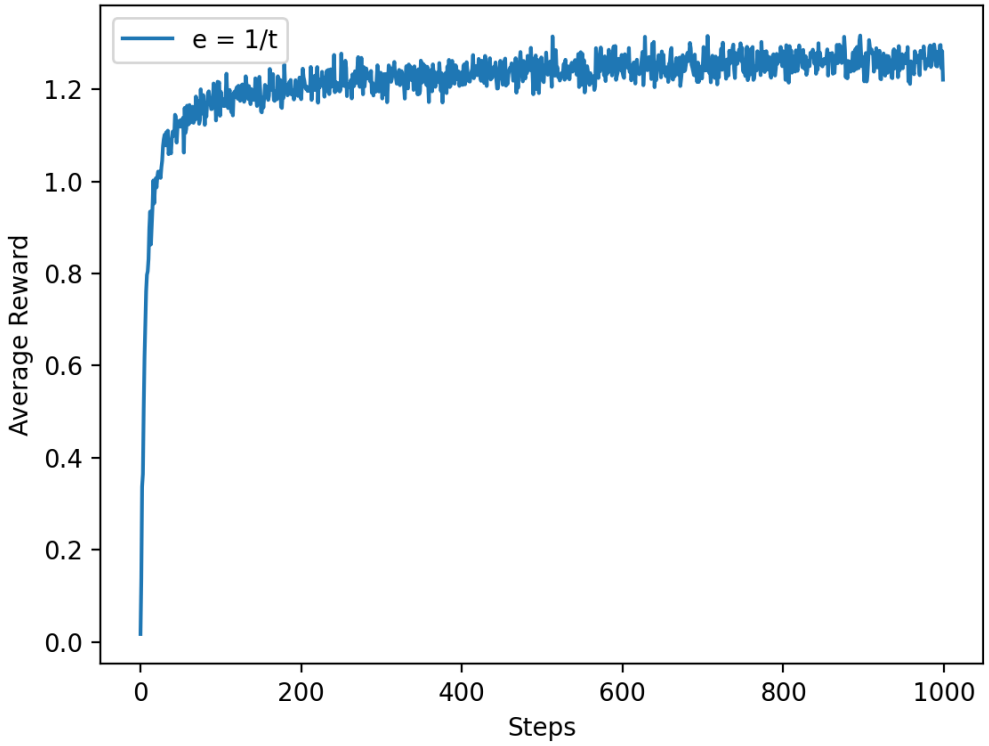

Epsilon = 0
Arm# 0
Arm# 1
Arm# 2
Arm# 3
Arm# 4
Arm# 5
Arm# 6
Arm# 7
Arm# 8
Arm# 9
Average Absolute Error
Steps
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0
200
400
600
800
1000

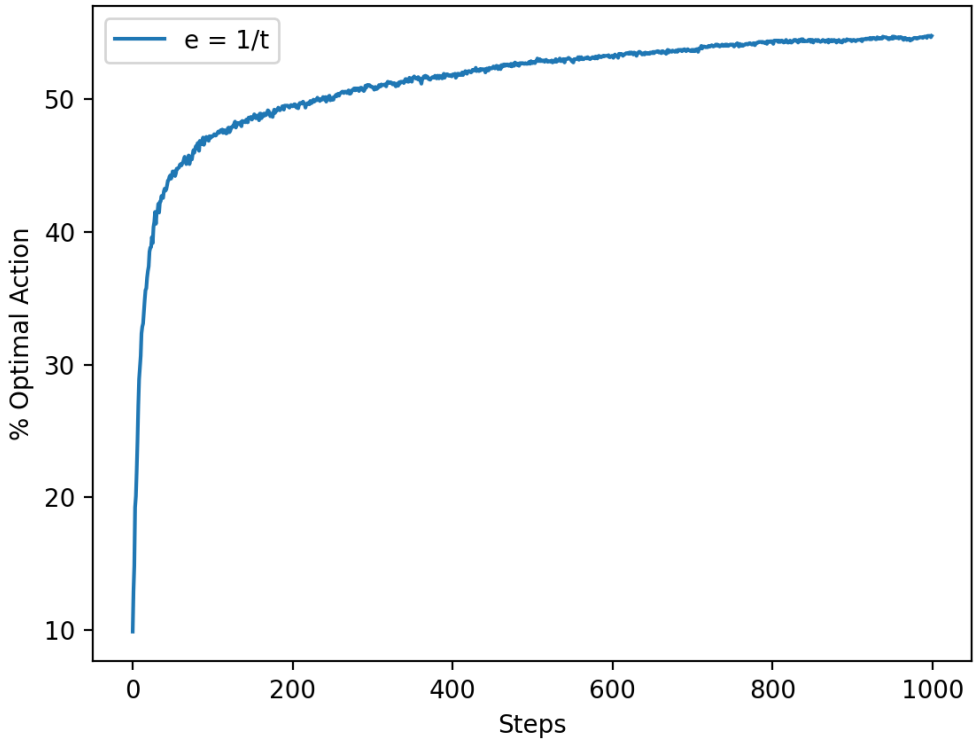

Epsilon = 0.1
Average Absolute Error
Steps
Arm# 0
Arm# 1
Arm# 2
Arm# 3
Arm# 4
Arm# 5
Arm# 6
Arm# 7
Arm# 8
Arm# 9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
200
400
600
800
1000

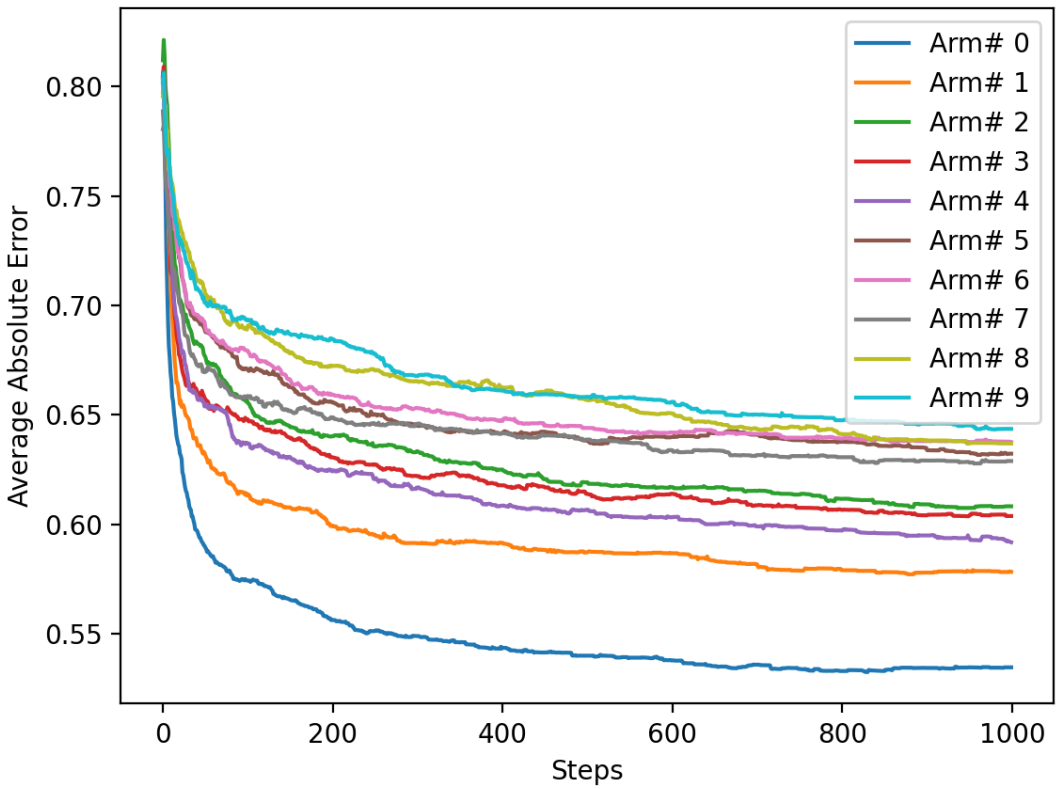

Epsilon = 0.01
Average Absolute Error
Steps
Arm# 0
Arm# 1
Arm# 2
Arm# 3
Arm# 4
Arm# 5
Arm# 6
Arm# 7
Arm# 8
Arm# 9
0.8
0.7
0.6
0.5
0.4
0
200
400
600
800
1000

For epsilon = 1/t

https://github.com/arora-ansh/RL-M2021_Ansh_2019022/blob/main/HW1/Q1/Q1b.py



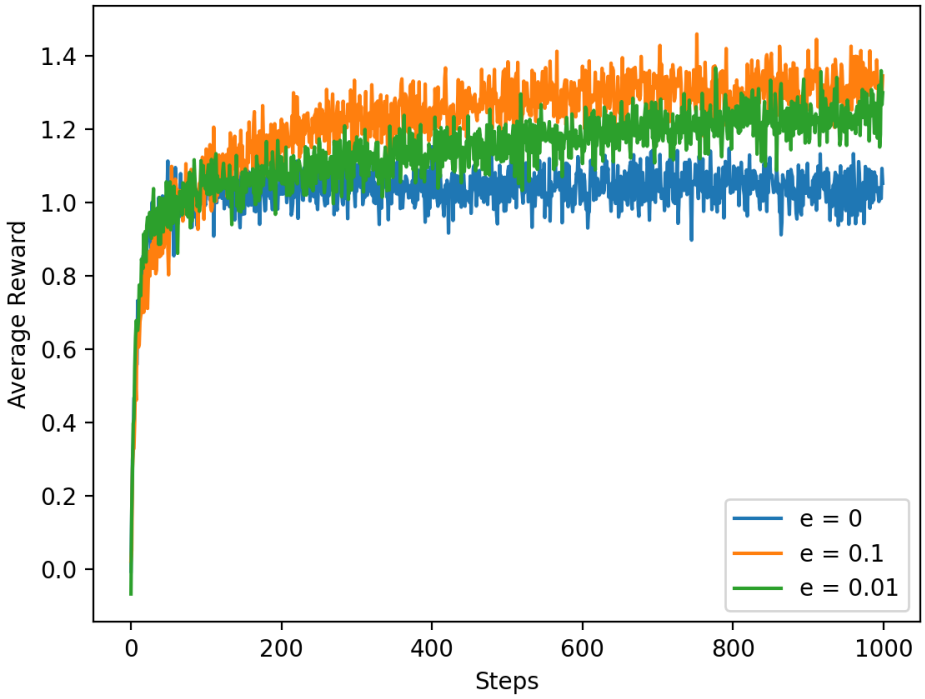


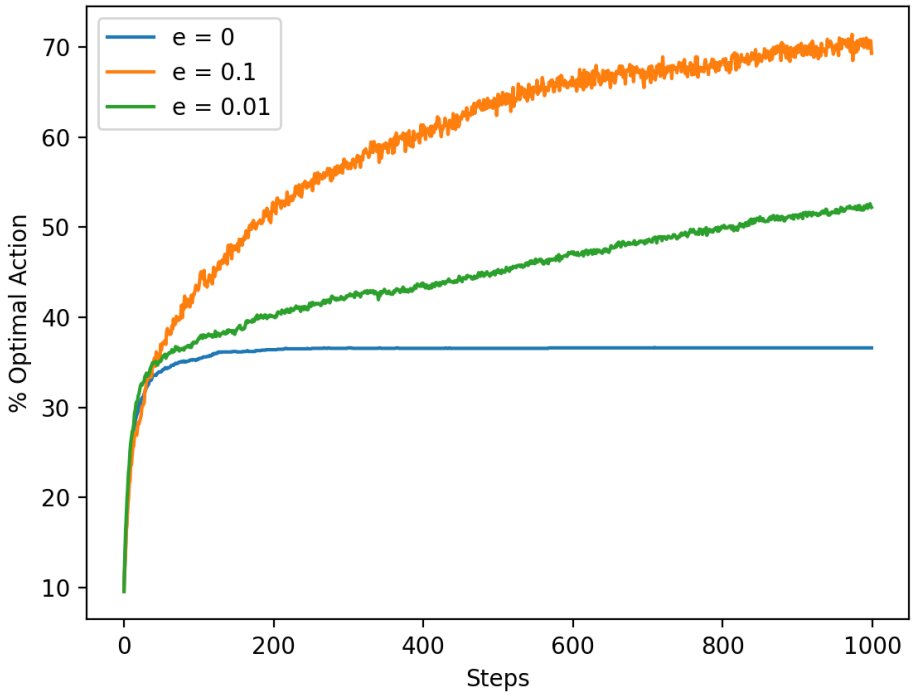

Arm# 0
Arm# 1
Arm# 2
Arm# 3
Arm# 4
Arm# 5
Arm# 6
Arm# 7
Arm# 8
Arm# 9
Average Absolute Error
0.80
0.75
0.70
0.65
0.60
0.55
0
200
400
600
800
1000
Steps

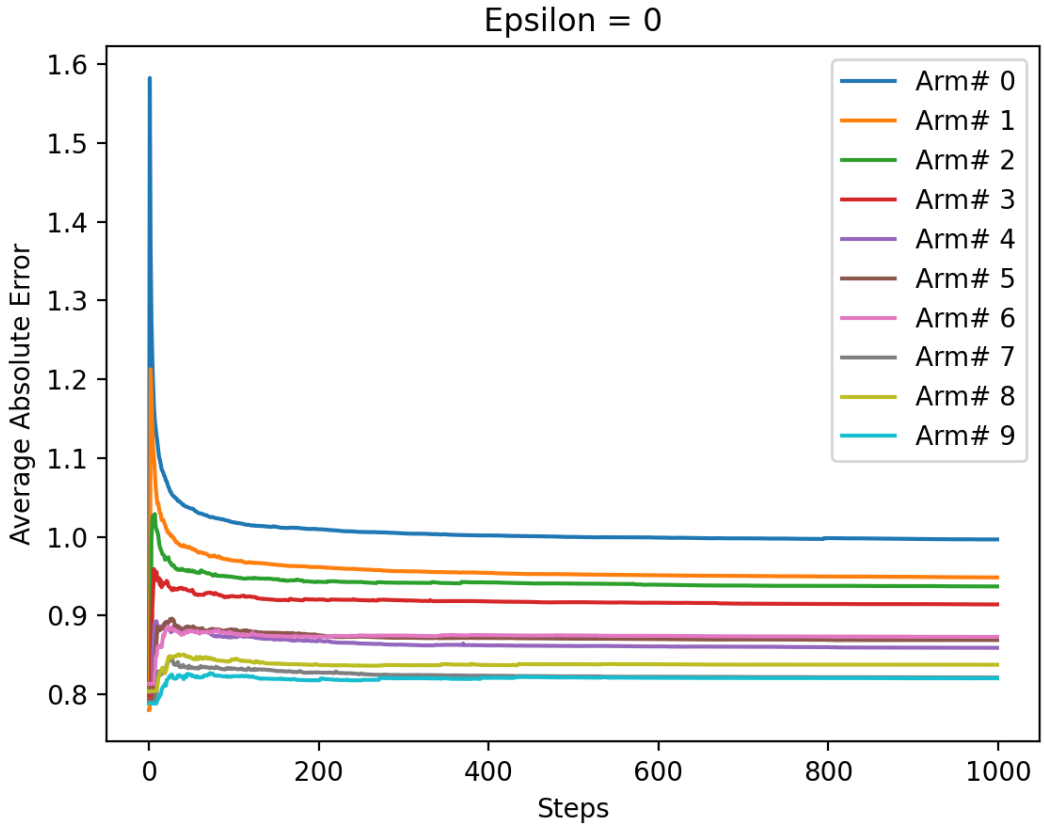
Question 2

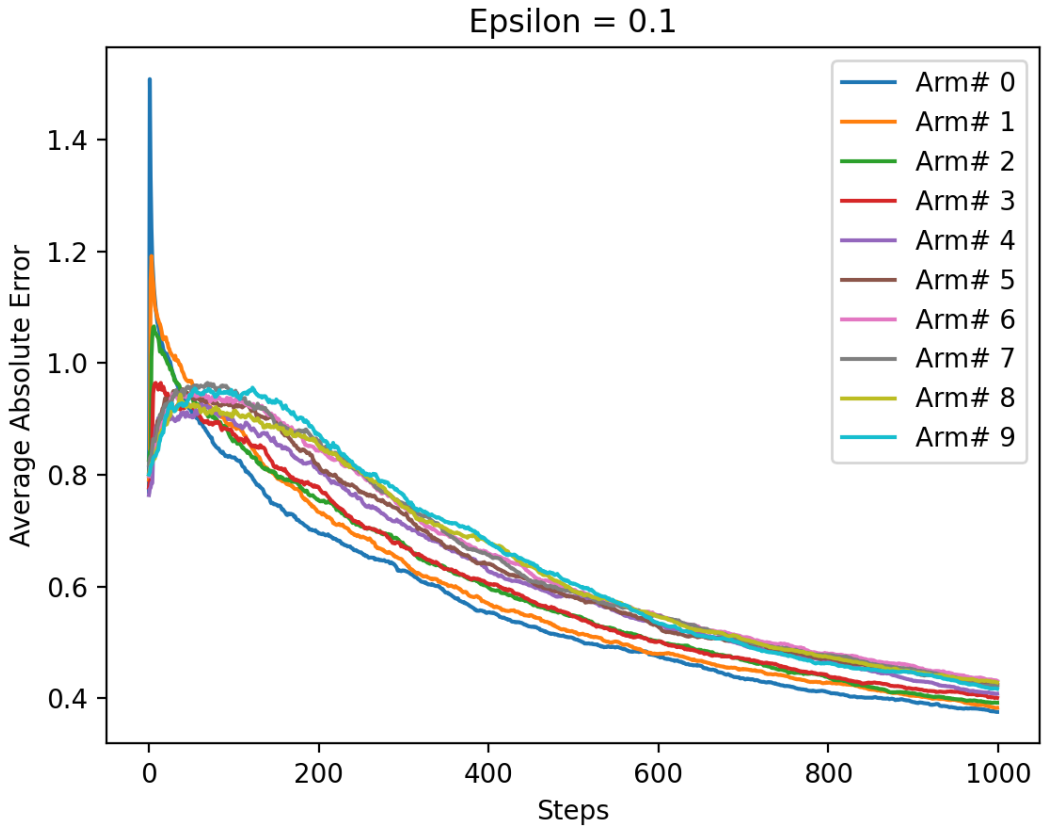
For epsilon = 0, 0.1, 0.01

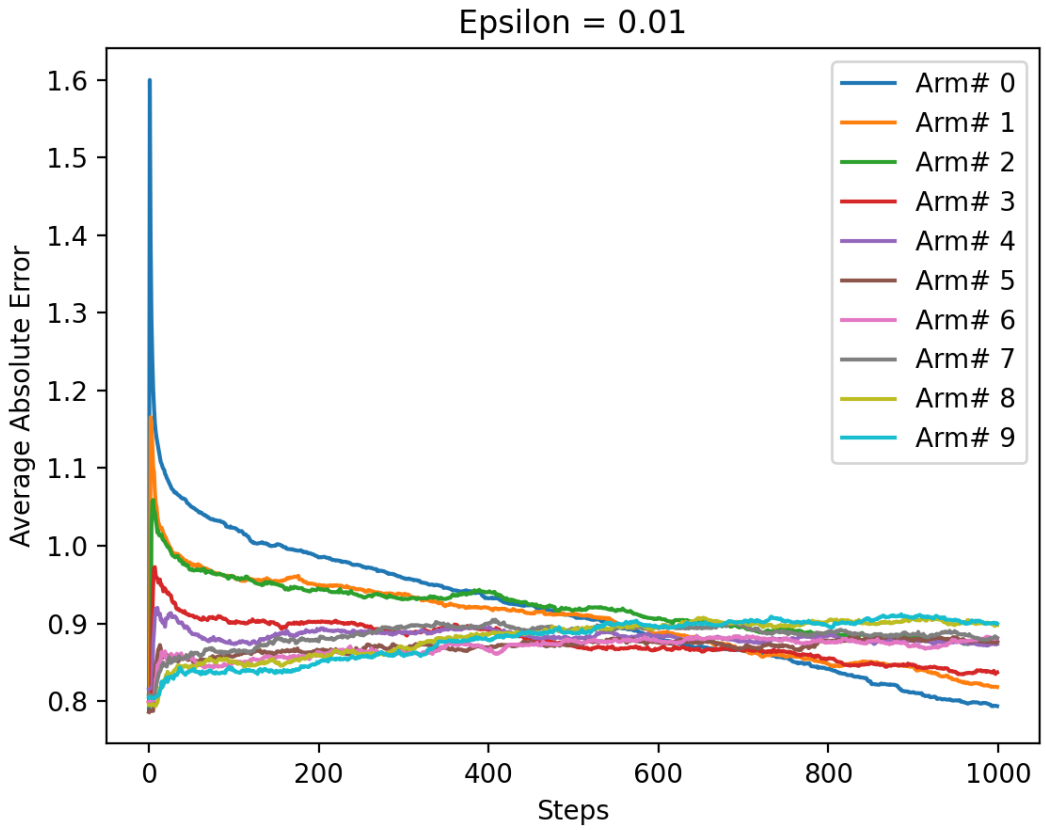
https://github.com/arora-ansh/RL-M2021_Ansh_2019022/blob/main/HW1/Q2/Q2.py





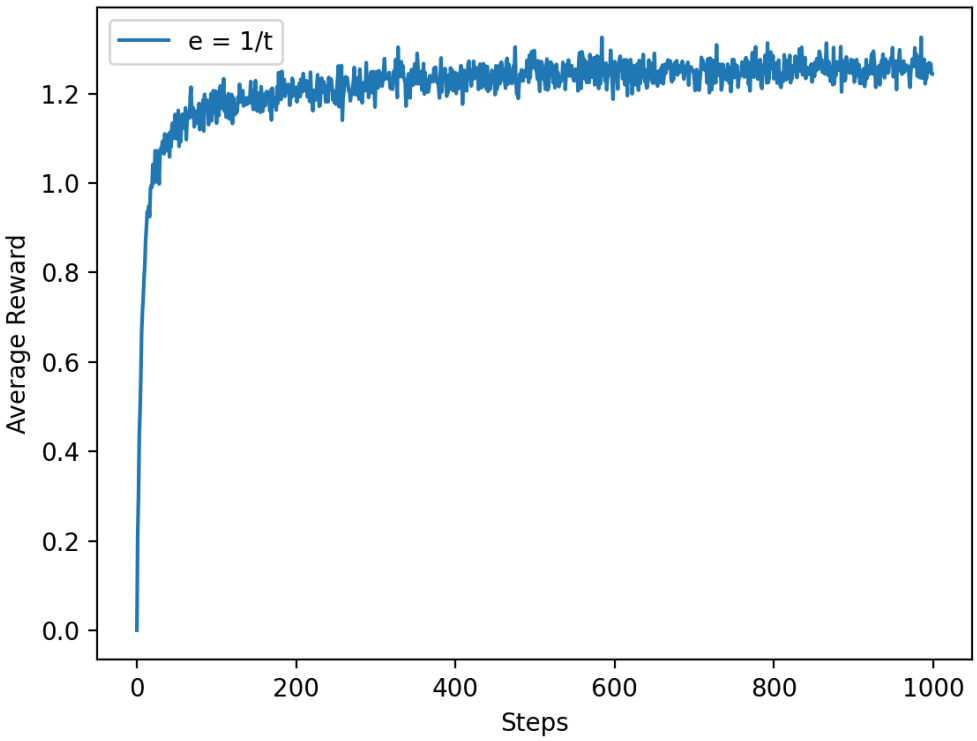

Epsilon = 0
Average Absolute Error
Steps
1.6
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0
200
400
600
800
1000
Arm# 0
Arm# 1
Arm# 2
Arm# 3
Arm# 4
Arm# 5
Arm# 6
Arm# 7
Arm# 8
Arm# 9

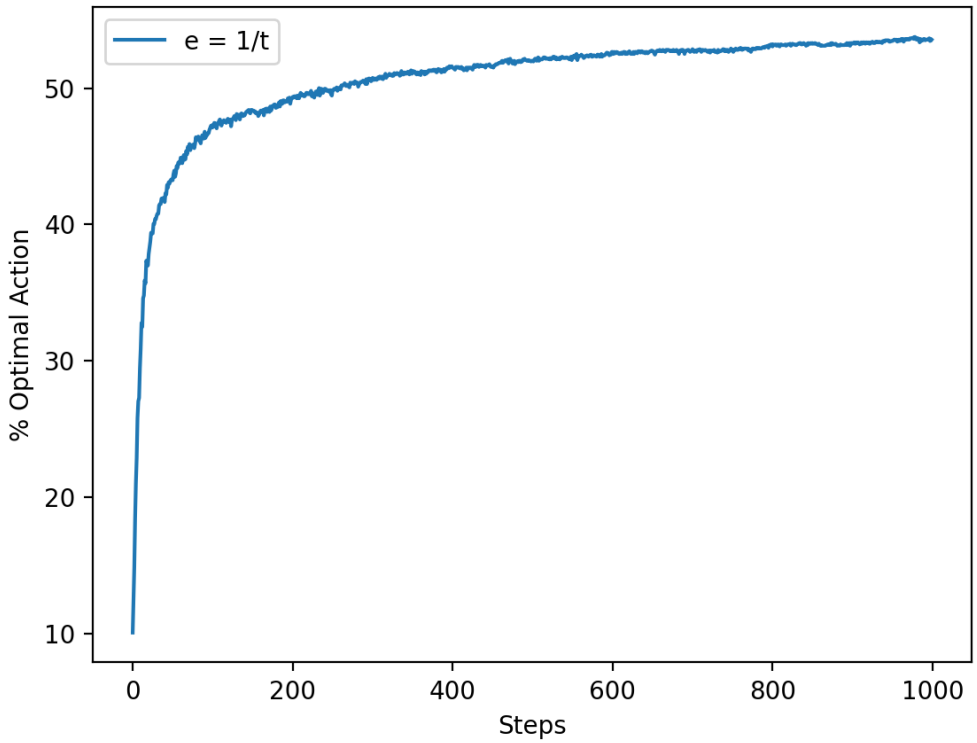

Epsilon = 0.1
Average Absolute Error
Steps
Arm# 0
Arm# 1
Arm# 2
Arm# 3
Arm# 4
Arm# 5
Arm# 6
Arm# 7
Arm# 8
Arm# 9
1.4
1.2
1.0
0.8
0.6
0.4
0
200
400
600
800
1000

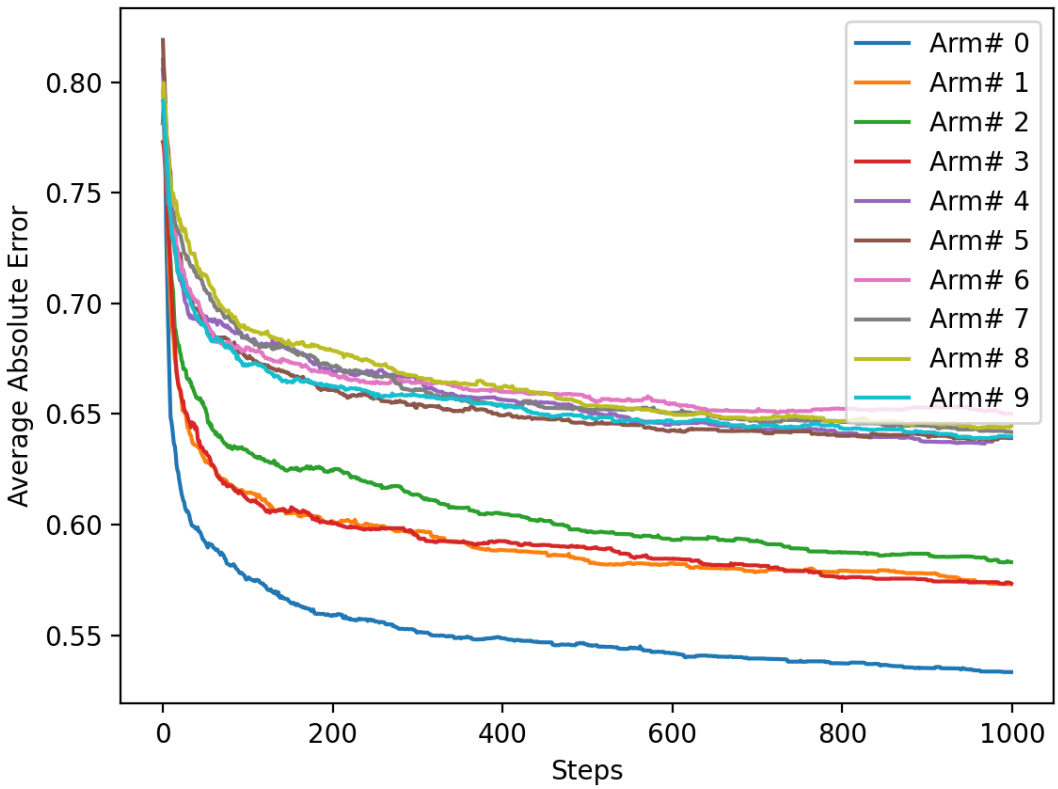

Epsilon = 0.01
Average Absolute Error
Steps
Arm# 0
Arm# 1
Arm# 2
Arm# 3
Arm# 4
Arm# 5
Arm# 6
Arm# 7
Arm# 8
Arm# 9
1.6
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0
200
400
600
800
1000

For epsilon = 1/t –

https://github.com/arora-ansh/RL-M2021_Ansh_2019022/blob/main/HW1/Q2/Q2b.py

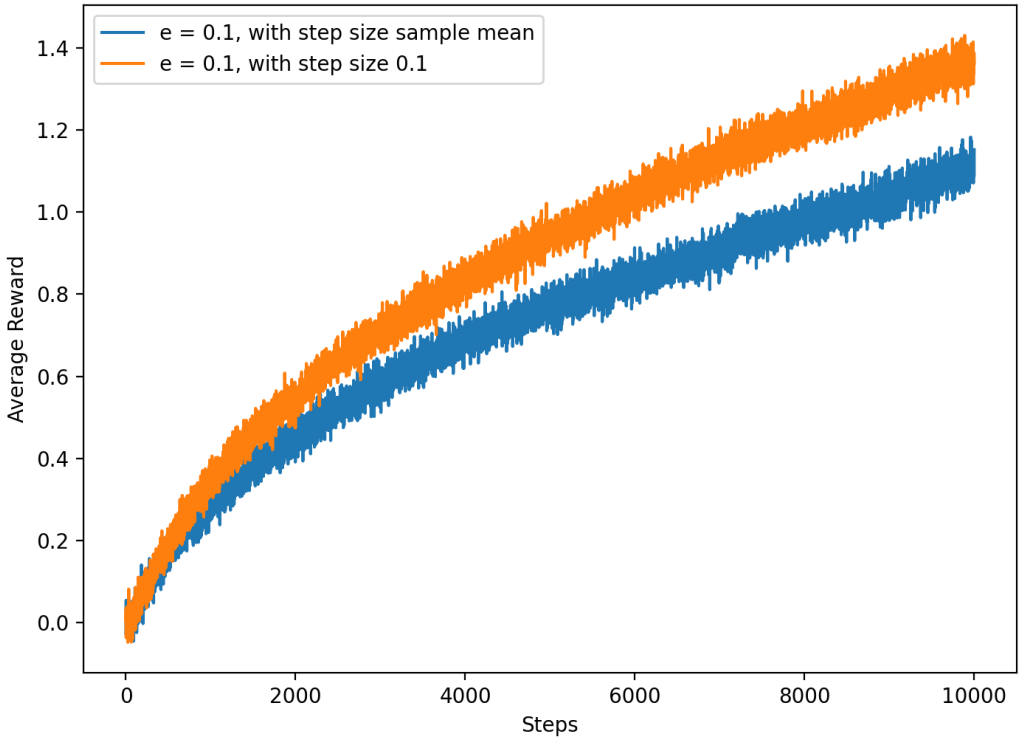


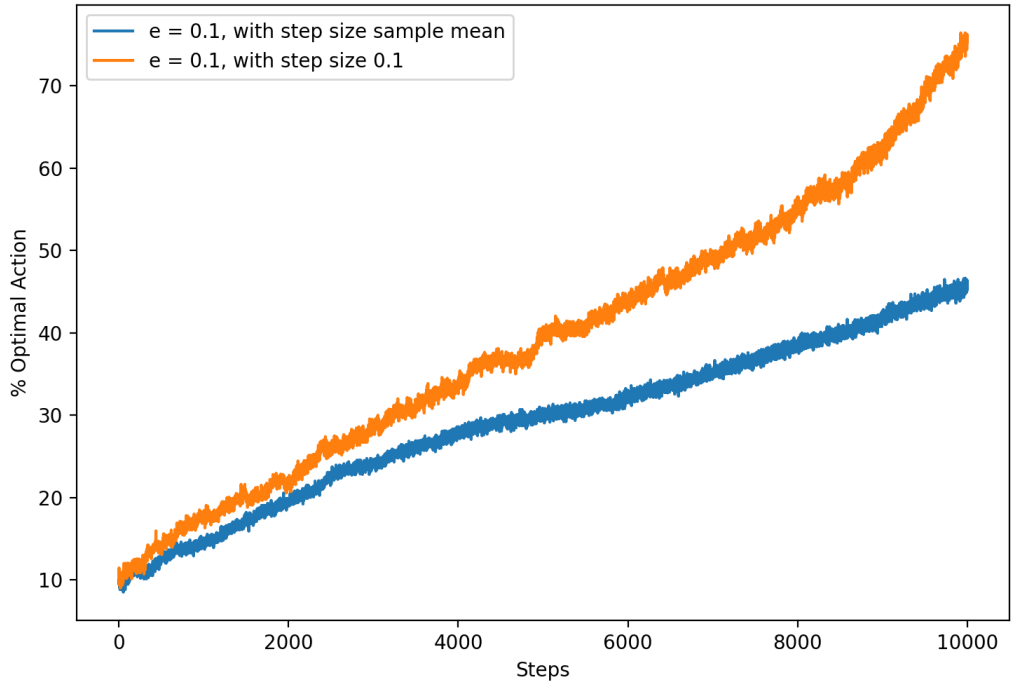



Arm# 0
Arm# 1
Arm# 2
Arm# 3
Arm# 4
Arm# 5
Arm# 6
Arm# 7
Arm# 8
Arm# 9
Average Absolute Error
0.80
0.75
0.70
0.65
0.60
0.55
0
200
400
600
800
1000
Steps

Question 5 -

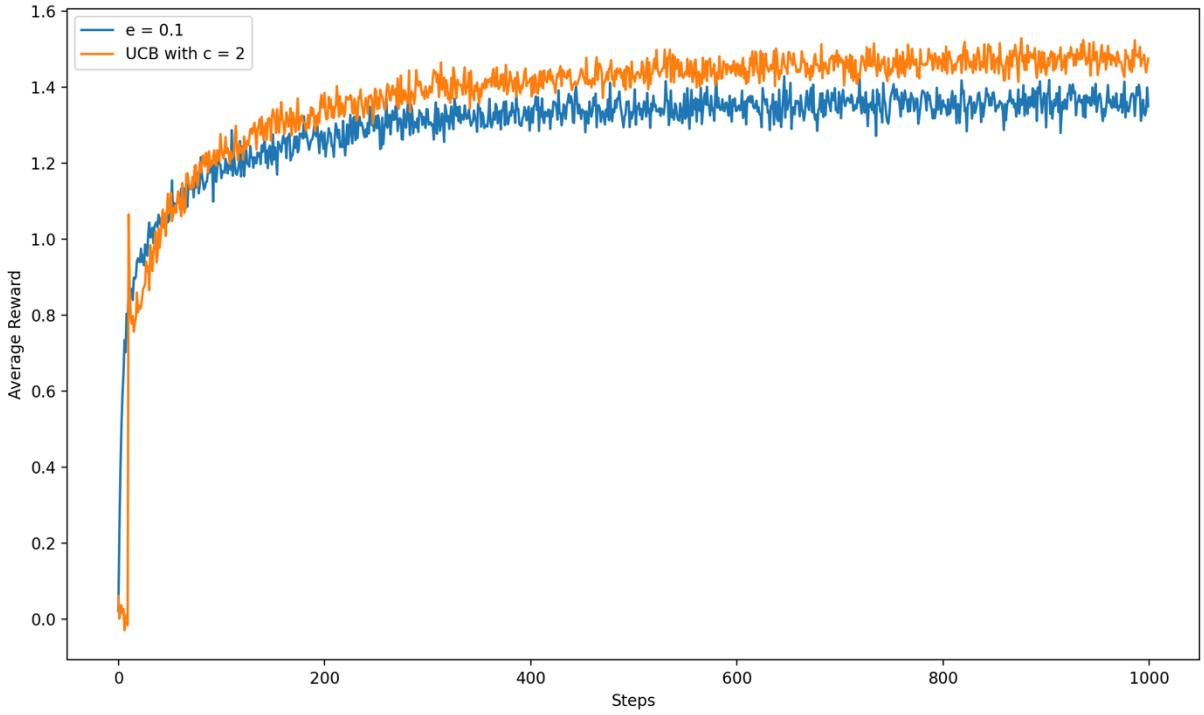
https://github.com/arora-ansh/RL-M2021_Ansh_2019022/blob/main/HW1/Q5/Q5.py





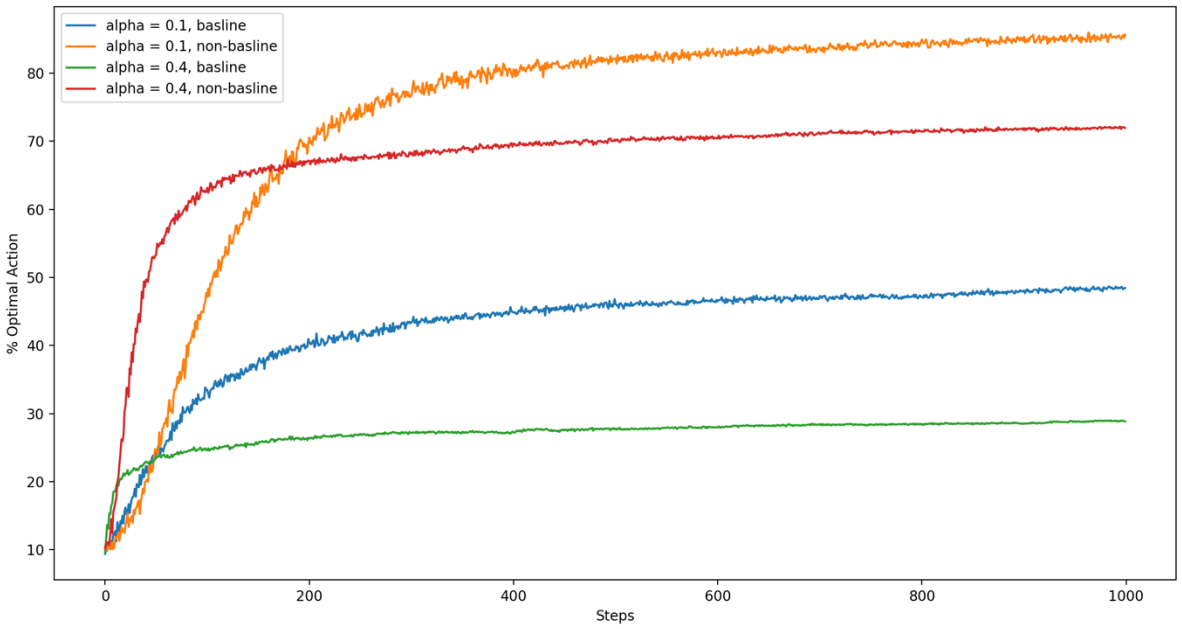
Question 6 -

https://github.com/arora-ansh/RL-M2021_Ansh_2019022/blob/main/HW1/Q6/Q6.py



Question 7 -

https://github.com/arora-ansh/RL-M2021_Ansh_2019022/blob/main/HW1/Q7/q7.py



RL - Reinforcement Learning.

Date: __/__/____

Ansh Arora 2019022

(3) Which method amongst $\varepsilon = 0.1$, $\varepsilon = 0.01$, $\varepsilon = 0$ and $\varepsilon = 1/t$ will perform the best.

For best performing, Expected value of that particular run should be highest. This would be highest when probability of picking the most optimal arm at infinity would be highest. Let this be $P(X)$.

1) $\varepsilon = 0.1$

$$P(X) = 1 - \varepsilon + \frac{\varepsilon}{K \text{ (no. of arms)}}$$

$$= 1 - 0.1 + \frac{0.1}{10}$$

$$= 0.9 + 0.01 = 0.901$$

2) $\varepsilon = 0.01$

$$P(X) = 1 - \varepsilon + \frac{\varepsilon}{K}$$

$$= 1 - 0.1 + \frac{0.01}{10}$$

$$= 0.99 + 0.001$$

$$= 0.991$$

3) $\varepsilon = 0$

$$P(X) = 1$$

However, in this particular case since there will be no exploration, there is a $1/10$ chance that we would be able to pick the right arm greedily making this inefficient.

Date: __/__/____

4) $\varepsilon = 1/t$

$$P(X) = \lim_{n \to \infty} 1 - \frac{1}{t} + \frac{1}{tk}$$

$$= 1$$

$\therefore$ $\varepsilon = 1/t$ would be the best performing method in the long run in terms of cumulative reword.

(4) In case of sample mean, after the action has been chosen once the dependence on initial choice goes away. This can be shown as follows -

$$Q_{t+1}(a) = Q_t(a) + \frac{1}{t}\left[R_t(a) - Q_t(a)\right]$$

At $t = 1$

$$Q_2(a) = Q_1(a) + \frac{1}{1}\left[R_1(a) - Q_1(a)\right]$$

$$\Rightarrow Q_2(a) = R_1(a)$$

Now,

At $t = 2$

$$Q_3(a) = Q_2(a) + \frac{1}{2}\left[R_2(a) - Q_2(a)\right]$$

$$= \frac{Q_2(a)}{2} + \frac{R_2(a)}{2}$$

and so on.

Date: __/__/____

since all future terms would now be dependent on Q_2, and Q_2 is independent of $Q_1(a)$, sample mean is not influenced by initial choice of $Q_1(a)$ $\forall a$.

Now, taking a constant stepsize α, we have

$$Q_{t+1}(a) = Q_t(a) + \alpha\left[R_t(a) - Q_t(a)\right]$$

$$= \alpha R_t(a) + (1-\alpha)Q_t(a)$$

$$= \alpha R_t(a) + (1-\alpha)\left[\alpha R_{t-1}(a) + (1-\alpha)Q_{t-1}(a)\right]$$

$$= \alpha R_t(a) + (1-\alpha)\alpha R_{t-1}(a) + (1-\alpha)^2 Q_{t-1}(a)$$

$$= (1-\alpha)^n Q_1(a) + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i$$

(2.6 from book)

Thus as can be seen above, it will be dependent on $Q_1(a)$, the initial choice $\forall a$.

For $\alpha \in (0, 1]$, if α value is reduced $(1-\alpha)$ increases, as a result of which the coefficient of initial choice $Q_1(a)$ $(1-\alpha)^n$ increases. Thus, the dependence is longer for a smaller α.

To deal with dependence on $Q_1(a)$, we can select the constants value to be 1.
This would mean $(1-\alpha)^n = 0^n = 0$ would be coefficient of $Q_1(a)$, thus making $Q_{t+1}(a)$ independent of $Q_1(a)$ $\forall a$.

Date __/__/____

(6) In the Upper Confidence Bound method,

$$A_t = \underset{a}{\operatorname{argmax}} \left[Q_t(a) + c \sqrt{\frac{\ln t}{N_t(a)}} \right]$$

Here $N_t(a)$ denotes the number of times that action a gets selected before time t. At $N_t(a) = 0$, a is considered to be a maximizing action.

In the first 10 steps, on an average the agent would cycle through all these actions (since it is taken to be maximal). On the 11th action, the agent will end up choosing greedily, leading to a sudden spike value.

Now, on the 12th step, $\sqrt{\frac{\ln t}{N_t(a)}}$ is equal to $\sqrt{\frac{\ln 12}{1}}$ for 9 of the 10 actions, while it is $\sqrt{\frac{\ln 12}{2}}$ for 1 of the actions that was picked greedily on the 11th step. Since this value is $\frac{1}{\sqrt{2}}$ times the last value that was picked on step 11, we can say that on average the pick would be lesser than step 11, hence leading to a fall in value.

∴ A spike is obtained on Step 11.