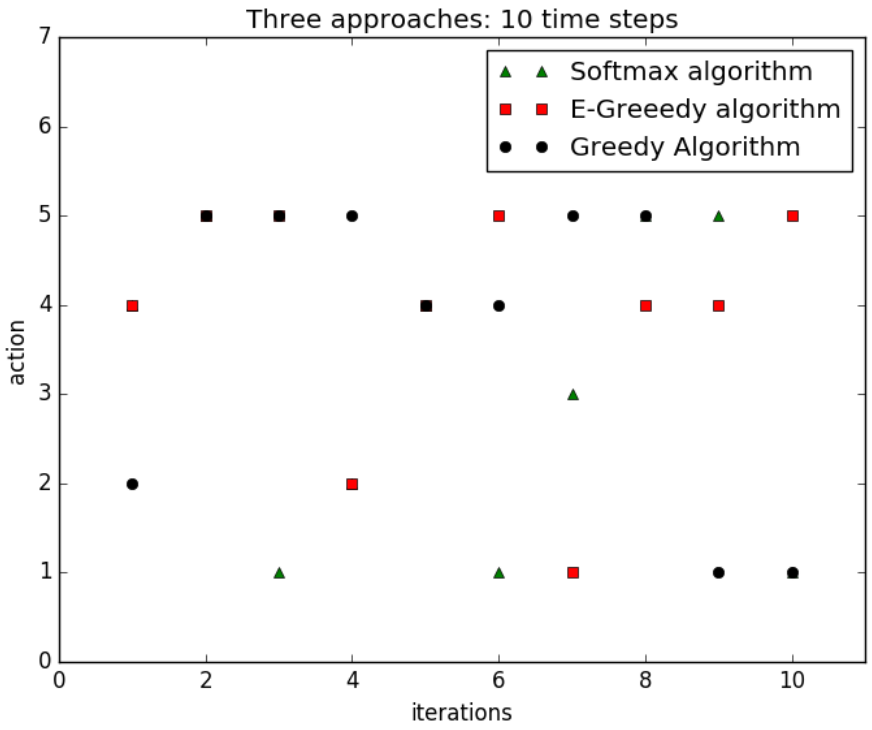
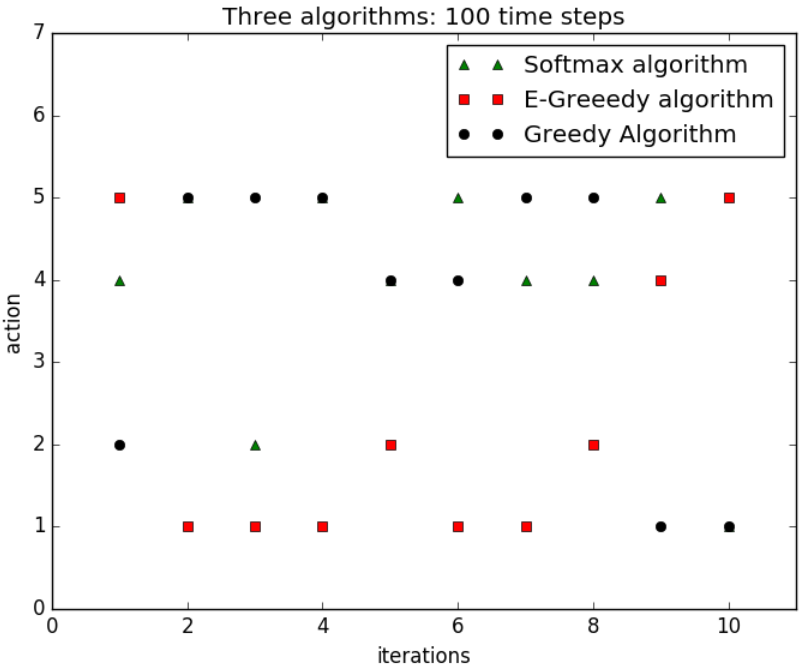
HOMEWORK 3
VENKATESH SRIPADA : 933271352

1. a) I have used the softmax, e-greedy and greedy approaches. When there are 10 time steps in the three approaches the results look scattered. There is no particular action to be taken (arm-pull).



This is because there is not enough time for the algorithm to learn and assign proper rewards. The average reward obtained per arm pull is 24.4358. The initial reward value is 50 for all.

b) When there are 100 time steps, the action performed is mostly 1 or 5. This is because there are more number of iterations to calculate the best action. There is high variance and mean for both these actions which makes it more probable to choose them as it is more likely to get a good reward.



In my simulations I have obtained an average reward of 5.18836. The initial reward was 50 for all actions. This means that there is more time to properly evaluate the rewards and get the best possible action. The algorithm constantly learns by iterating the rewards so that a constant reward can be given to the best action.

2. In the case of a 5x10 grid world, the agent reaches the goal, sometimes takes a long time and sometimes does not reach the goal in the given time. It necessarily does not take the shortest path even if it is very close to goal. If the random position is initialized close to the goal the agent reaches goal definitely but if it is in the beginning it takes close to 17-18 steps or may not reach the goal.

This is because it takes the small 'e' probability of moving to the state where the path is rerouted, hence longer to reach destination. The agent traverses in a loop, eventually not converging in the 20 steps.

3. The Q-learning algorithm performed better than the previous algorithm as it accommodated for the discount factor. The Q-value is initialized to be 50. Also the start location is at 3,5 which is close to the goal 4,10 and not a random start. This gives a constant and quick convergence.

It converged in a minimum of 6 steps and a maximum of 17 steps.

Shortest path: Step 1

```
[[-1. -1. -1. -1. -1. -1. -1. -1. -1. -1.],
 [-1. -1. -1. -1. -1. -1. -1. -1. -1. -1.],
 [-1. -1. -1. -1. -1. 39. -1. -1. -1. -1.],
 [-1. -1. -1. -1. -1. -1. -1. -1. -1. 100.],
 [-1. -1. -1. -1. -1. -1. -1. -1. -1. -1.]]
```

Step2:

```
[[-1. -1. -1. -1. -1. -1. -1. -1. -1. -1.],
 [-1. -1. -1. -1. -1. -1. -1. -1. -1. -1.],
 [-1. -1. -1. -1. -1. 39. 39. -1. -1. -1.],
 [-1. -1. -1. -1. -1. -1. -1. -1. -1. 100.],
 [-1. -1. -1. -1. -1. -1. -1. -1. -1. -1.]]
```

Step3:

```
[[-1. -1. -1. -1. -1. -1. -1. -1. -1. -1.],
 [-1. -1. -1. -1. -1. -1. -1. -1. -1. -1.],
 [-1. -1. -1. -1. -1. 39. 39. 39. -1. -1.],
 [-1. -1. -1. -1. -1. -1. -1. -1. -1. 100.],
 [-1. -1. -1. -1. -1. -1. -1. -1. -1. -1.]]
```

Step4:
[[-1. -1. -1. -1. -1. -1. -1. -1. -1. -1.],
 [-1. -1. -1. -1. -1. -1. -1. -1. -1. -1.],
 [-1. -1. -1. -1. -1. 39. 39. 39. 39. -1.],
 [-1. -1. -1. -1. -1. -1. -1. -1. -1. 100.],
 [-1. -1. -1. -1. -1. -1. -1. -1. -1. -1.]]

Step5:
[[-1. -1. -1. -1. -1. -1. -1. -1. -1. -1.],
 [-1. -1. -1. -1. -1. -1. -1. -1. -1. -1.],
 [-1. -1. -1. -1. -1. 39. 39. 39. 39. 39.],
 [-1. -1. -1. -1. -1. -1. -1. -1. -1. 100.],
 [-1. -1. -1. -1. -1. -1. -1. -1. -1. -1.]]

Step6:
[[-1. -1. -1. -1. -1. -1. -1. -1. -1. -1.],
 [-1. -1. -1. -1. -1. -1. -1. -1. -1. -1.],
 [-1. -1. -1. -1. -1. 39. 39. 39. 39. 39.],
 [-1. -1. -1. -1. -1. -1. -1. -1. -1. 79.],
 [-1. -1. -1. -1. -1. -1. -1. -1. -1. -1.]]

It takes more number of steps when it moves left (away from goal) in the first step and traverses all the way to 3,0 and comes back.

The key difference was that the agent knew its current state and hence estimated the future states based on that. The Q table gives the agent a better idea of the path to be taken.

Hence it is a optimal for the agent to know its current state and estimate the future states while performing a go-to-goal problem. It can be implied that the agent does not need to know the whole environment before it starts exploring. Also, the state in which agent is initialized does not make a difference if it there are n-time steps.