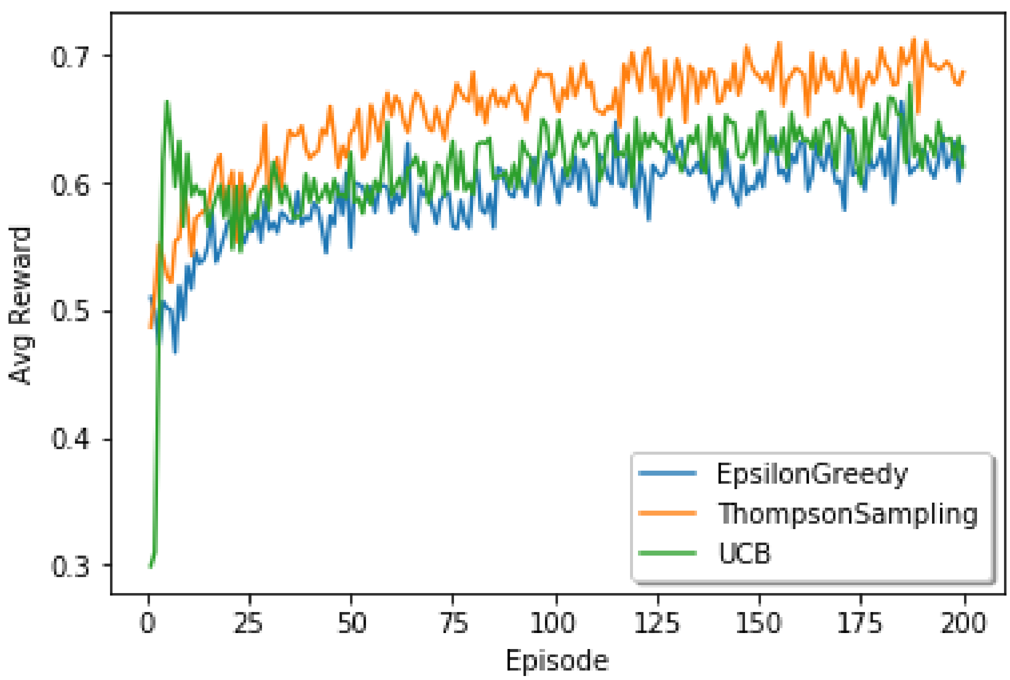
Assignment2 Q1

June 27, 2018

```
In [1]: %matplotlib inline
```

```
In [2]: %run TestBandit.py
```

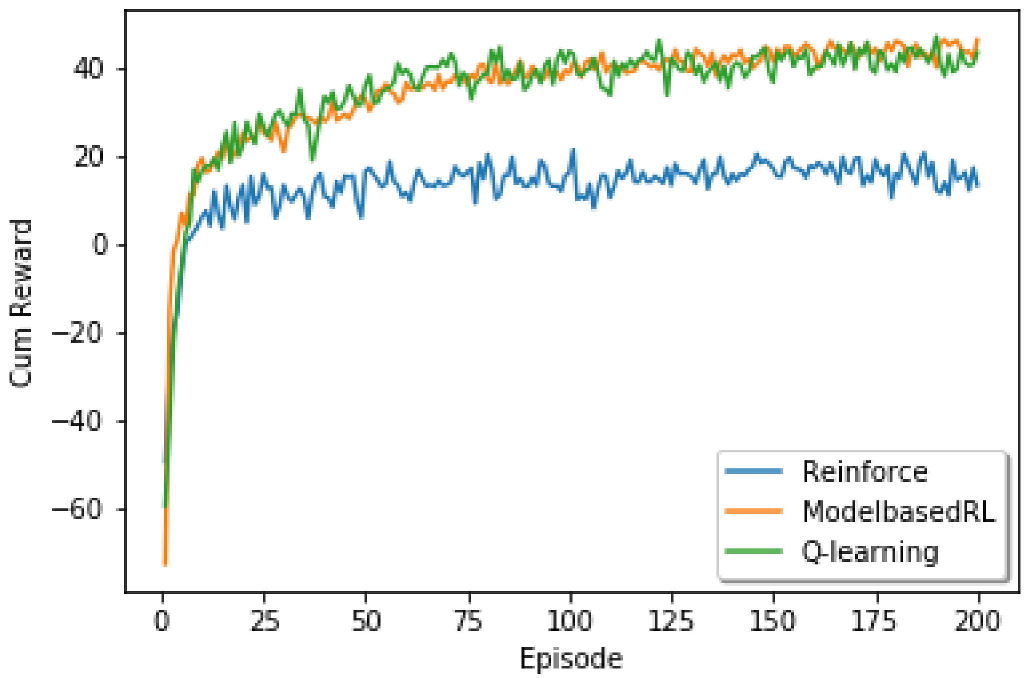
```
C:\Users\Lin Daiwei\Assignment2\RL2.py:222: RuntimeWarning: divide by zero encountered in log
  ucb = empiricalMeans + np.sqrt( 2*np.log(i)/n_action)
C:\Users\Lin Daiwei\Assignment2\RL2.py:222: RuntimeWarning: invalid value encountered in sqrt
  ucb = empiricalMeans + np.sqrt( 2*np.log(i)/n_action)
```



From the graph, we can see that Thompson sampling has highest average rewards. It is because we set k = 1 and this gives a very high probability of exploration at the begining. When episodes increases, the distribution becomes peaked and the exploration decreases. The UCB performs relatively better than epsilon greedy method, which is expected.

```
In [3]: %run TestRL2Maze.py
```

```
trial  0
trial  10
trial  20
trial  30
trial  40
trial  50
trial  60
trial  70
trial  80
trial  90
```



From the graph, we can see that model-based RL and Q-learning has very similar performance. Notice that the epsilon is set to 0.3 in model-based RL, while in Q-learning, it is 0.05. It means Q-learning needs less exploration than model-based RL given same number of episodes.

In addition, they both achieve higher rewards than REINFORCE algorithm. The reasons why Reinforce method gives lower rewards are as follows: 1. the action selection is stochastic. When an action is the best among all actions, the sampleSoftmaxPolicy() will not always return the optimal action. Different from the exploration in model-based RL and Q-learning, the chance of taking non-optimal actions are much higher. This will reduce the average reward obtained. 2. Here we use fixed learning rate in REINFORCE, and this will increase variance, especially after some knowledge is gained. However, if we use 1/n(s,a) as learning rate, the REINFORCE cannot learn useful information.