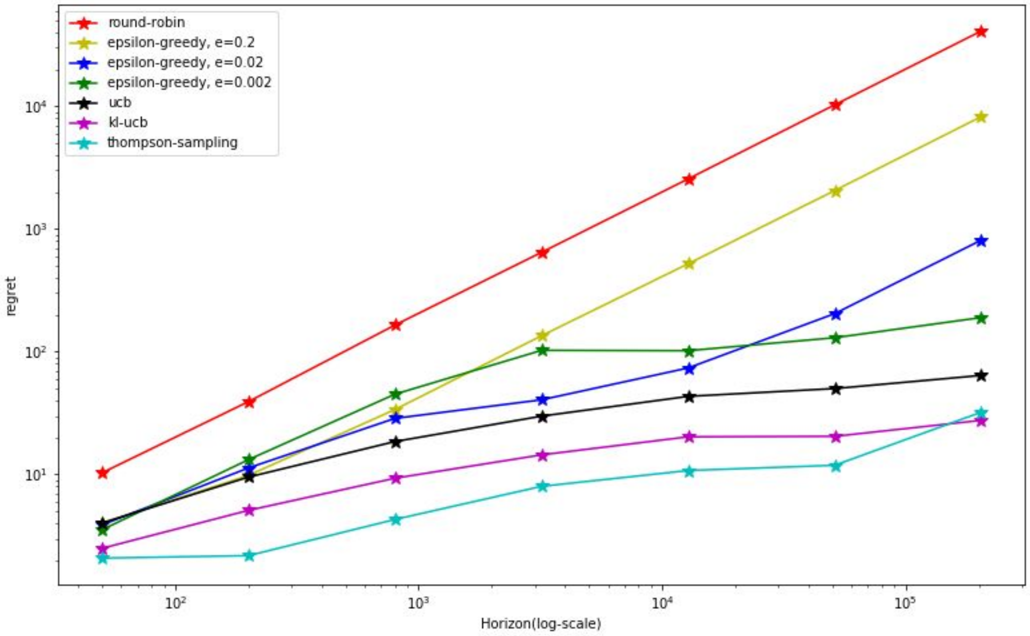
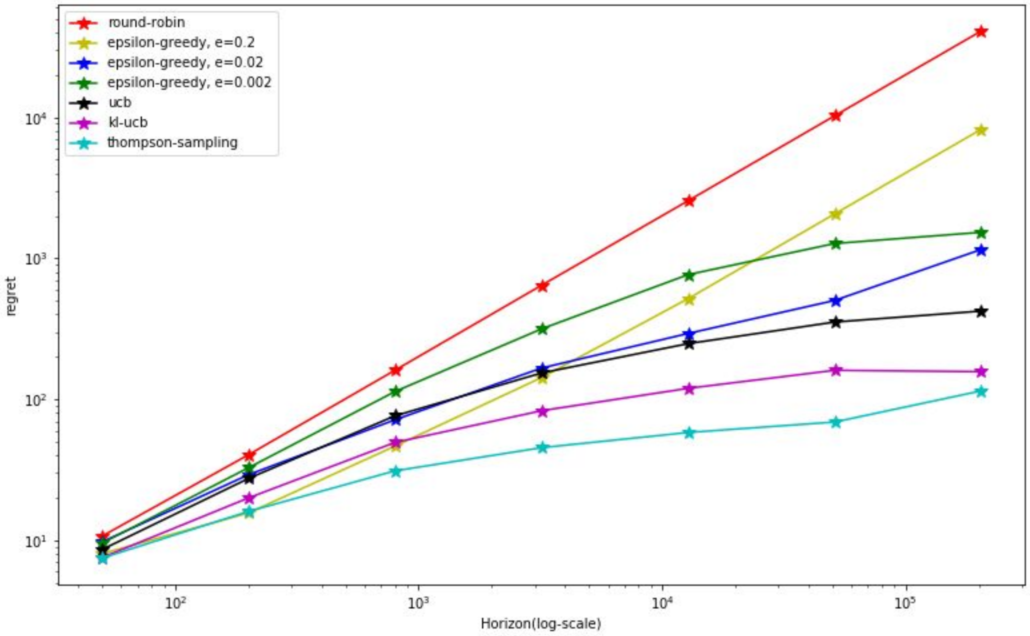
CS747-Assignment-1

SUDHIR KUMAR SUMAN
16D070027

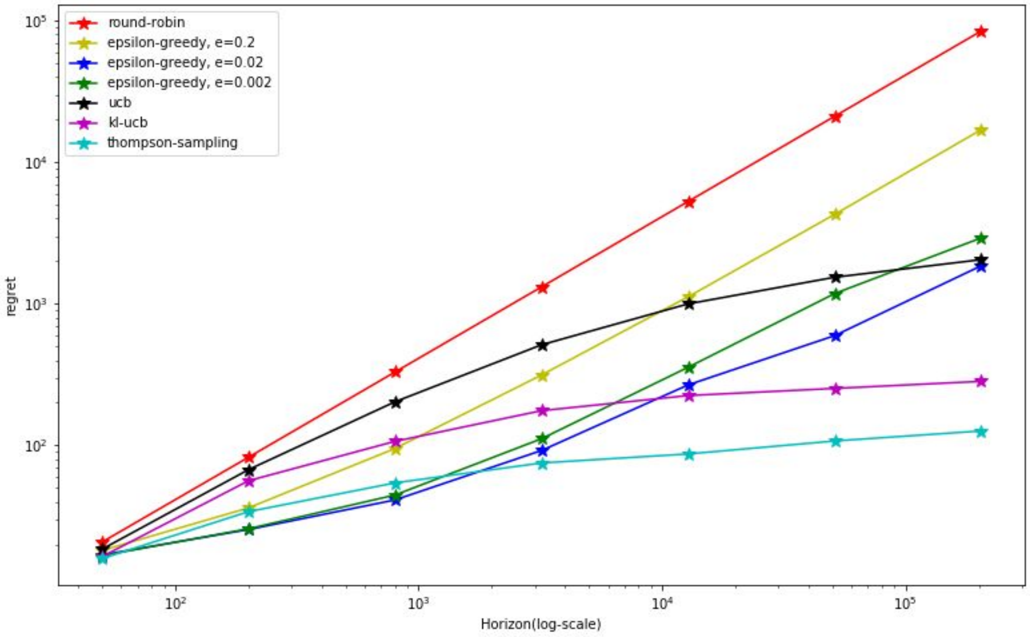
Plot of different Algorithm on instance_1:



Plot of different Algorithm on instance_2:



Plot of different Algorithm on instance_3:



Interpretation:

Here we can see from the graph that epsilon greedy with epsilon = 0.2 is not working well on higher horizon as it is exploring too much instead of exploiting the arm with higher reward. It is working like round-robin when epsilon value is high.When horizon is large than epsilon greedy with lower epsilon will work better

Round-Robin algorithm is working like a linear function and it is getting worse when the horizon increases.

As expected Thompson Sampling is working better than all the algorithm.

The performance of algorithm on instances for higher value of horizon are in the following order:
Thompson_Sampling > KL-UCB > UCB > epsilon_greedy (e = 0.002) > epsilon_greedy (e = 0.02) > epsilon_greedy (e = 0.2) > Round-Robin

Algorithm Assumption:

1. **Round Robin:**
   Pulled arms uniformly horizon number of times

2. **Epsilon-Greedy:**
   Assumption:
   For the first n steps I only explored and after that I exploit and explored both as per the epsilon greedy algorithm.

3. **UCB:**
   Assumption:
   For the first n steps I only explored all the arms in order to get some values to compute bound and later

4. **KL-UCB:**
   Assumption:
   When the reward is zero than I added 0.0001 to the reward in order to avoid zero in log function and when reward is 1 than directly took of q = 1. Also the maximum error in value of q can be upto 0.01.

5. **Thompson Sampling:**
   Assumption:
   At the very first,I selected the arm randomly and afterward based on Thompson Sampling algorithm as the probability of selecting all arms will be equal in very first case as per the beta equation and argmax will give only the first element everytime to make it indeterministic I also randomized even very first select for each horizon .

***n is number of arms**