REPORT **170050067**

CODE:

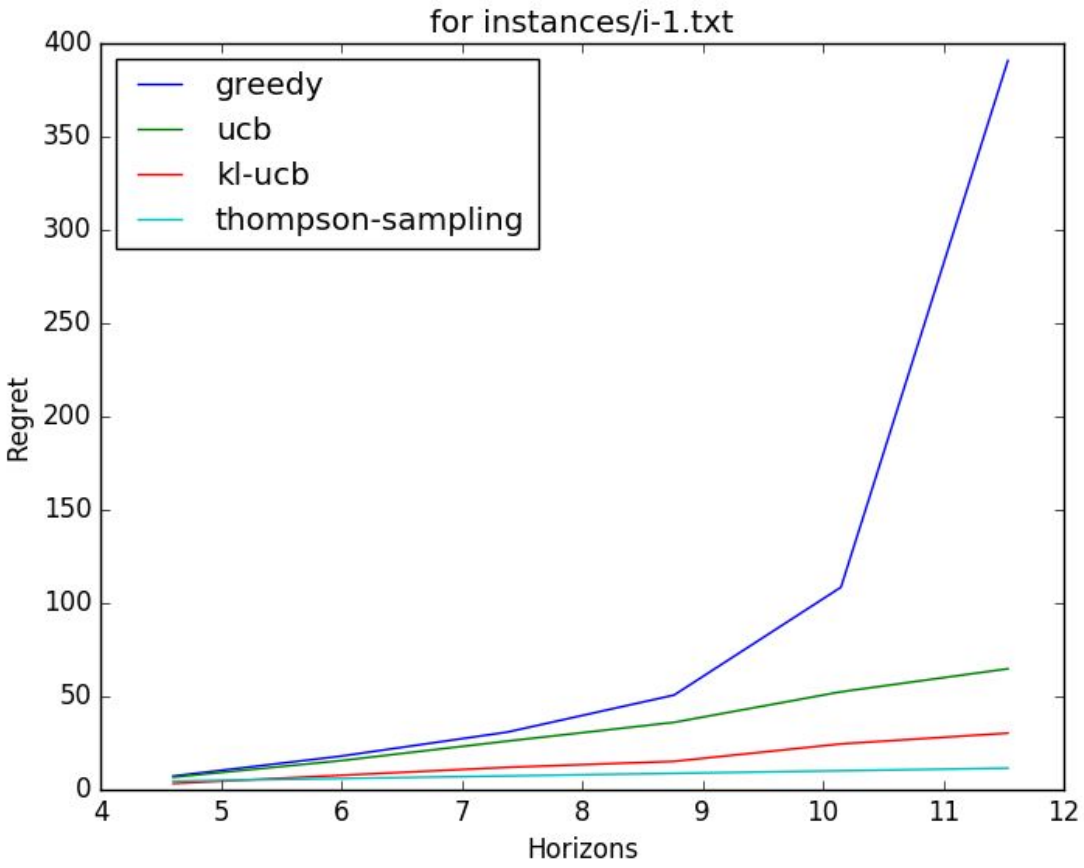
Written code for each algorithm in separate files and imported them into bandit.py

Task-1

assumptions:

For greedy,ucb and kcb

Initial pulled all the arms initials ones



Interpretation: for all the three instances together.

1.Epsilon greedy Given epsilon=0.01

As the number of pulls increases,in epsilon we are going to explore even after we have found The highest mean,hence epsilon greedy would always be having the greater regret compared to other while horizon reaches infinity

2.Similarly Comparing UCB and KL-ucb,

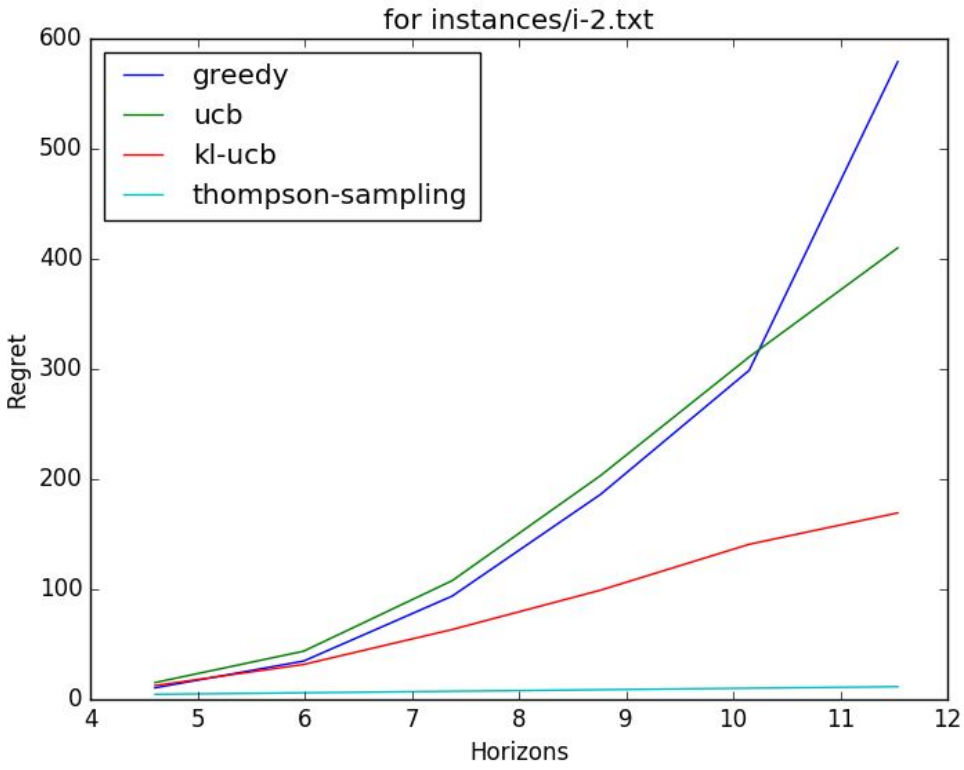
Regret of ucb > KL-ucb as there is a Lai and Robbins lower bound ,and hence KL-UCB has lesser Confidence interval while compared to UCB.

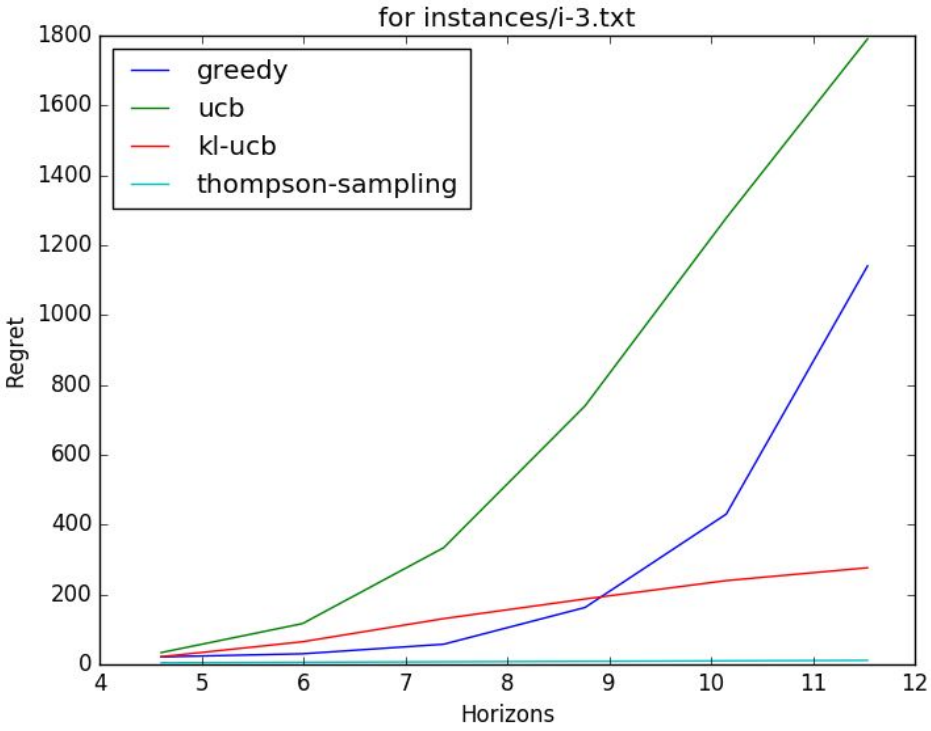
3.Thompson Sampling:this has least regret of all

Initial we are taking the prior from uniform distribution,and calculating the posterior for each Based on the reward,hence this converges to a beta distribution.from bayesian inference.

Hence we are always calculating the expected mean based all the previous pulls,rewards

So it has a lower regret,Compared to remaining.





The KL-ucb,are straight lines as regret is of order(log(t)) T is horizon.implementation of algo is written in end of report.

Task-2

I have used the given sorted true means,and calculated the next arm to be pulled using the discrete distribution.

In thomson-sampling as we dont have the true means,we pick them from the uniform distribution,in hint we are given the true mean

Algorithm:

Create n beliefs for n arms initially

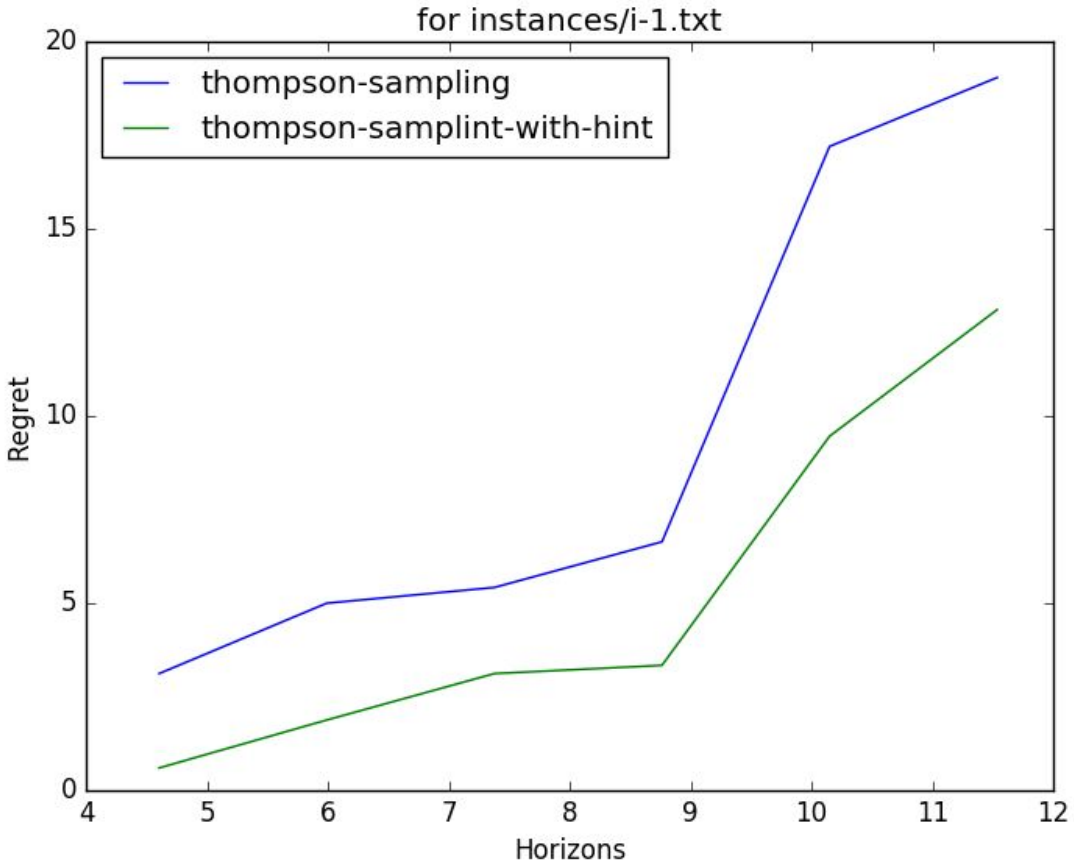
Pulled the arm which has the highest belief for the largest true mean [(argmax(beliefs.truemean)]

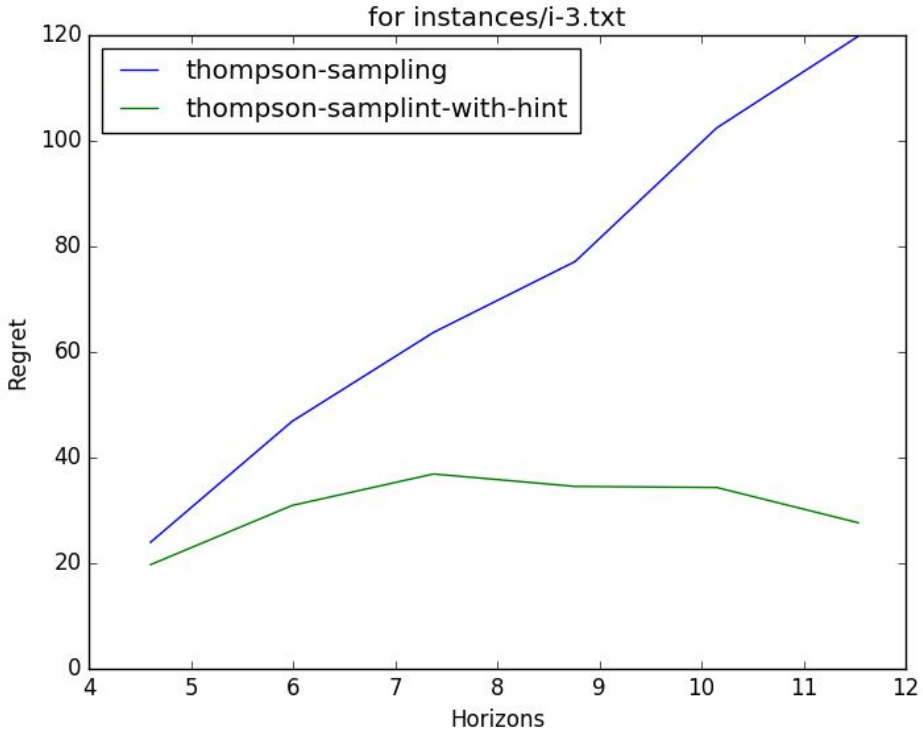
Calculated the posterior for the arm based on the reward ,updated the belief of pulled arm

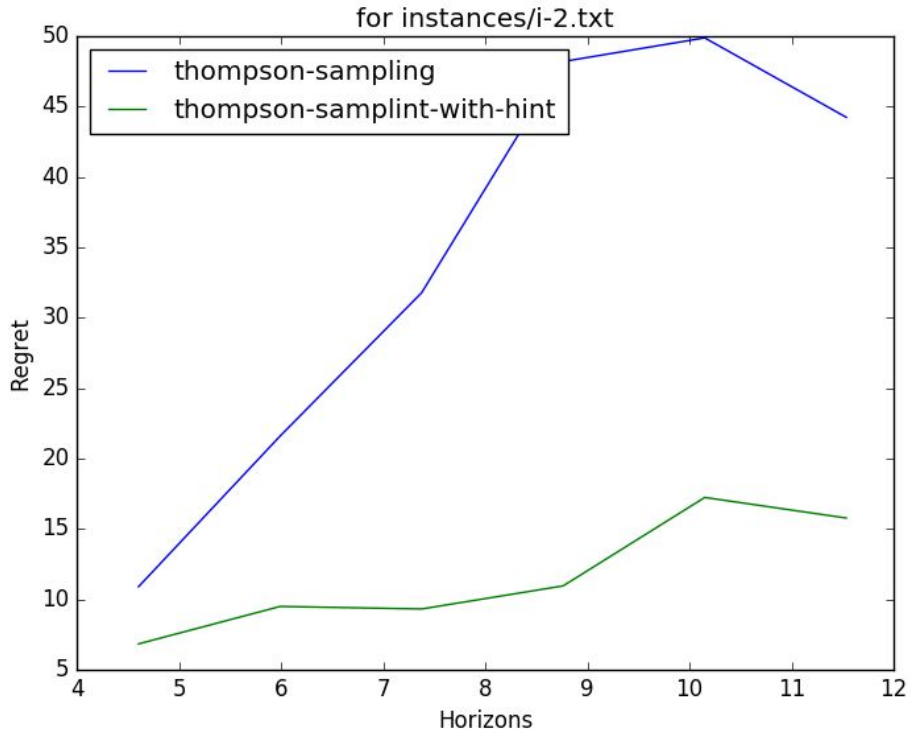
Problems faced:

Initial i was choosing the optimal arm by purely taking the discrete distribution and pick the arm with highest mean,extraced from np.random.choice(truemeans,1,p=beliefs[arm])

Here this was better compared to other but has become mostly random,hence picked only the beliefs of highest true mean beliefs later,that gave good results







Horizons are of ln scale.

TASK-3

Experimental Results:

For, take the average REG of 50 random seeds (0 through 49) for the horizon of 102400.

instance-1 regret[0.001,0.01,0.1]=[347.44, 255.68, 2043.1]

instance-2 regret[0.001,0.01,0.1]=[1847.6, 511.28, 2078.76]

instance-3 regret[0.001,0.01,0.1]=[2001.4, 1085.32, 4295.9]

We could see the regret for each instance with epsilon 0.01 is less compare to epsilon of 0.001 and 0.1

This because difference number of exploraes and exploits
If epsilon=0.001 explores are more
If e=0.1 exploits are more
Hence there is always a e2, where e1<e2<e3 where regret of e2 is less
Due to difference of number exploit and explores for a given bandit.

Implementation of each algorithm:

All the algo are written in python class,one for each algo and are imported in bandit.py
Each algo has following functions:

- Init
- Preprocess
- Pullarm
- getReward
- Updatemean
- getRegret

Epsilon-GreedyAlgo:

In pull arm function returned a particular arm based on random number generated,if number less than epsilon,return random arm else argmax(mean).

UCB:

Calculated the confidence interval for each arm in pullarm function for each pull,and returned the arm with Highest pa+sqrt(2*ln(t)/ut)

KL-ucb:

Calculated q for each arm such q=[pa,1] , ua*KL(pa,q)<=log(t)+3*log(log(t))
Here used precision for finding q that is 0.001 and c=3,as in algo given c>=3
Returned arm with max q

ThomsponSampling:

Calculated the random sample from beta(sa+1,fa+1) for each arm,
Returned the arm with max(sample)