REPORT

OF

PROGRAMMING ASSIGNMENT 1

CS 747

NAME : *ARPIT DWIVEDI*
ROLL NUMBER: *200100032*

TASK 1:

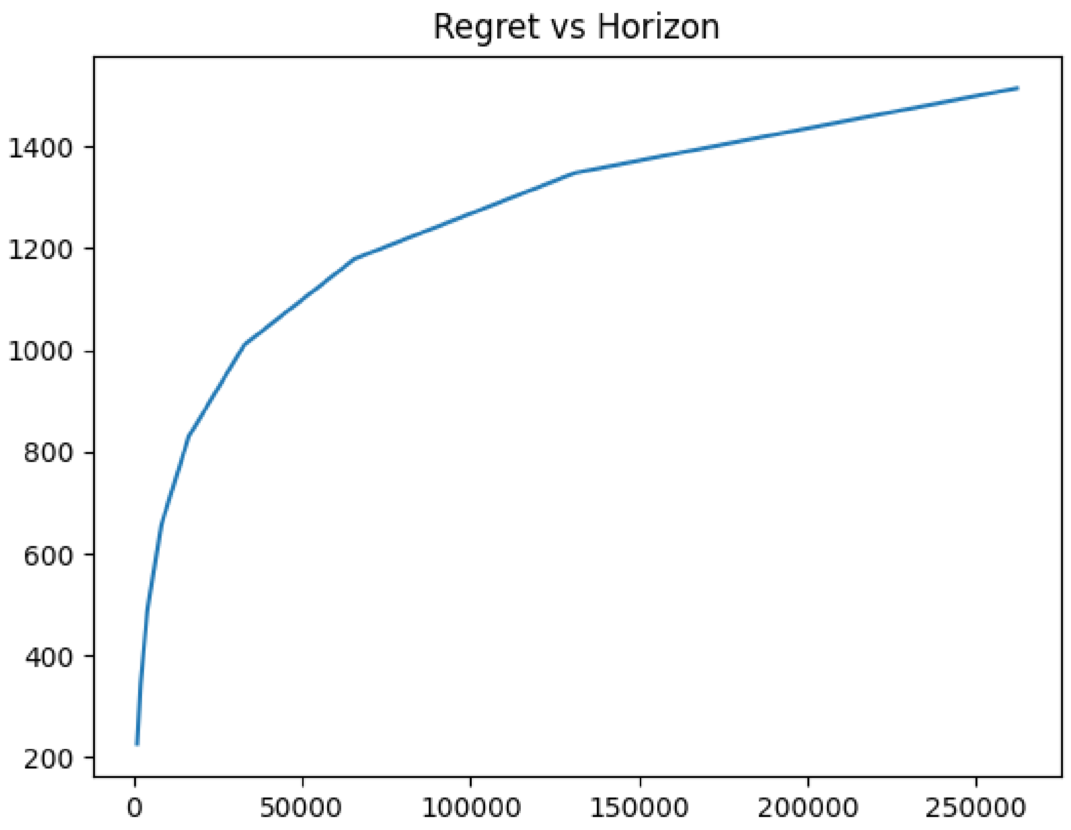
UCB-Aglorithm:

The structure of algorithm is as:

- 'Step.check' := a variable used to check whether before we began to choose arms deterministically by taking a maximum of UCB bound among all arms, all arms have been explored at least once.

- Till 'srep.check = 0' round robin exploration continues
- The moment we reach the other end of the arms index(if we give them indexing), UCB for all arms has been calculated and stored in the 'self.ucb_values' variable.

- Then we choose the arm with maximum UCB value and return it as the value of give_pull function.
- If the round robin exploration has been finished then only we begin the UCB calculation that is ensured with an if statement in code before calculation.
- The ‘self.values[i]’:={empirical reward} of previous step(t-1) has been used for calculation of UCB(at t_th step of choosing arm) as clearly stated in the definition of Algorithm in slides.
- We can see the sub-linear regret for the case
- Following is the plot obtained:



KL_UCB Algorithm:

The structure of the algorithm is as:

- Round robin exploration before kl_ucb calculation begins
- Two helper function used for calculation of kl_ucb value according to definition provided in slides
- FIRST: The evaluation of expression for

u*KL_UCB - ln(t) - cln(ln(t)):

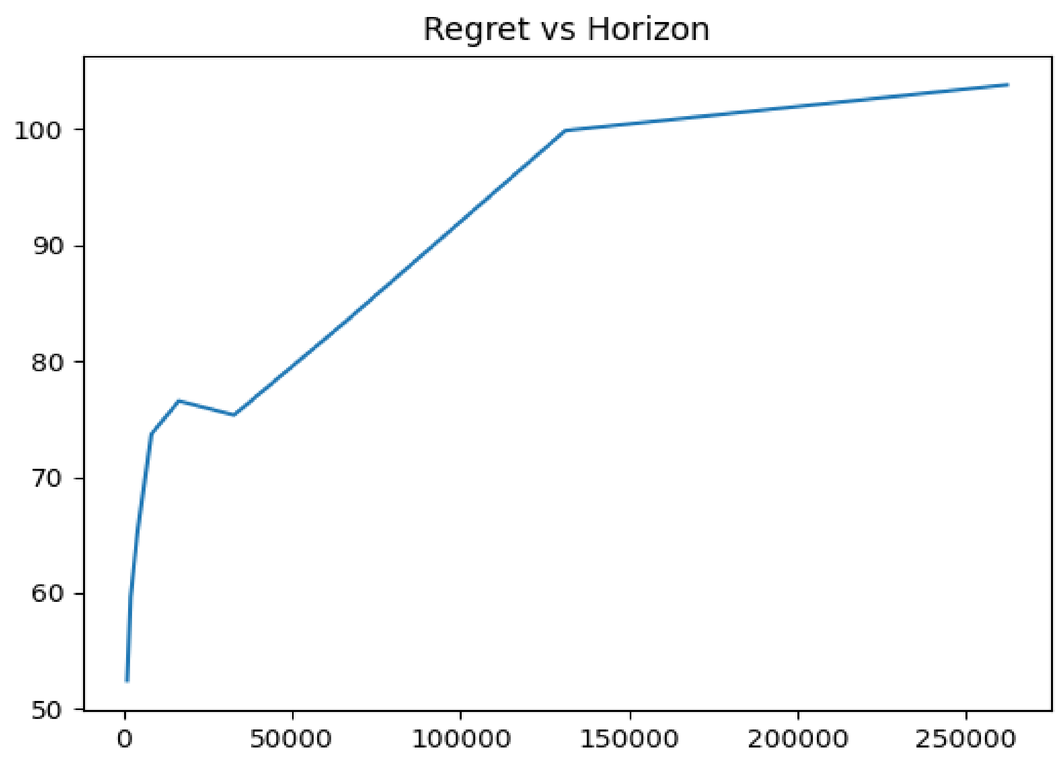
1. CASE 1 when p = 0 we need to consider only second term in KL_divergence expression
2. CASE 2 when p = 1 we need to consider only first term in KL_divergence expression

- The bisection method has been used for calculation the q value till the interval in which root is found become less than 1e-6
- Then as usual the arm index with maximum kl_ucb has been returned for called give_pull function
- Following is the regret i am getting, i am not attaching the plots as it takes a lot of time to run the kl_ucb for
    - TESTCASE 1: 119.2

Thompson Sampling:

The structure of the algorithm is as:

- The inbuilt beta distribution function is directly used for obtaining the beliefs of arms
- Then storing the number of success and failures in corresponding variables we can easily get the belief.
- We can see the sub-linear regret here
- Following is the plot obtained:

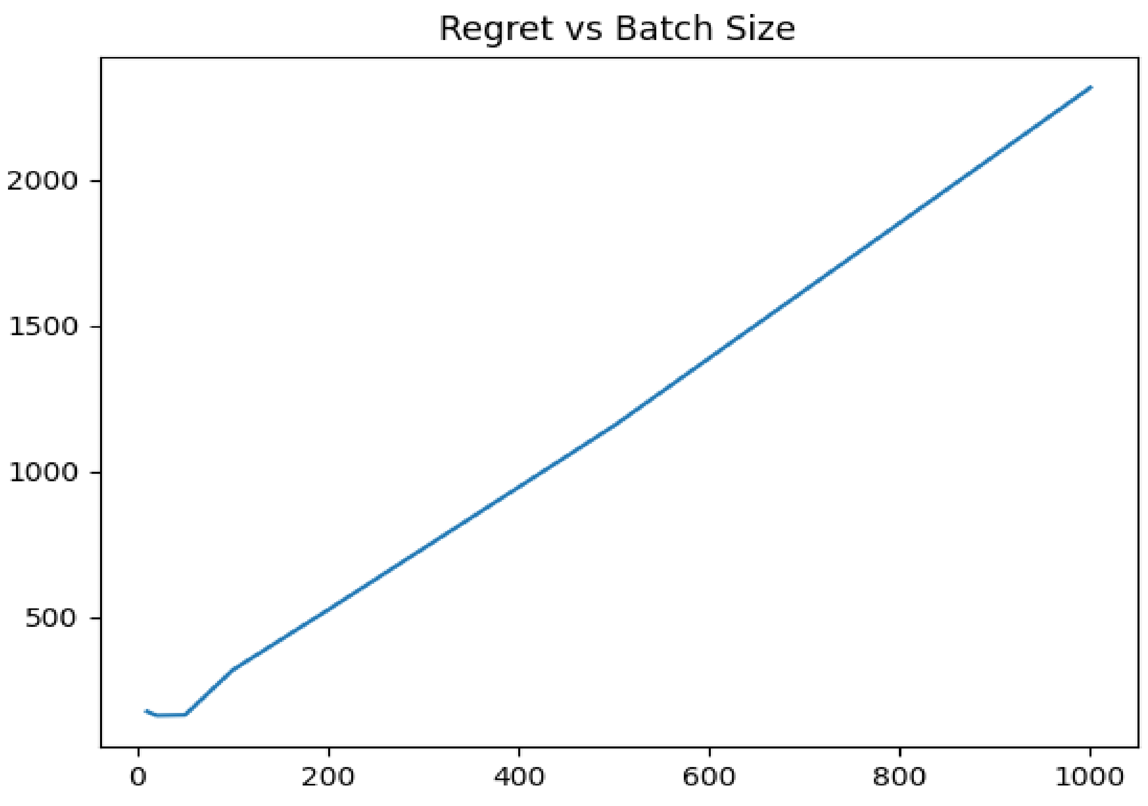


TASK 2:

The structure of code is as follows:

- The first give pull call are totally exploring all the arms until all the values in 'self.counts' are at least 1
- This is bifurcated in 3 cases :
  - FIRST (number of arms < batch size): We will pull all arms at least once and then randomly pull any arms till total pulls equal to batch size
  - SECOND (number of arms = batch size) : We will pull each arms once
  - THIRD (number of arms > batch size) : We will choose first batch size number of arms and pull all of them once and pass the rest of the unexplored arms to an variable 'self.arms_indexes_left' and when next time the give pull is called first it is checked that whether all the arms are explored which is False in this case hence the above three conditions will be checked till all the arms are explored at least once.

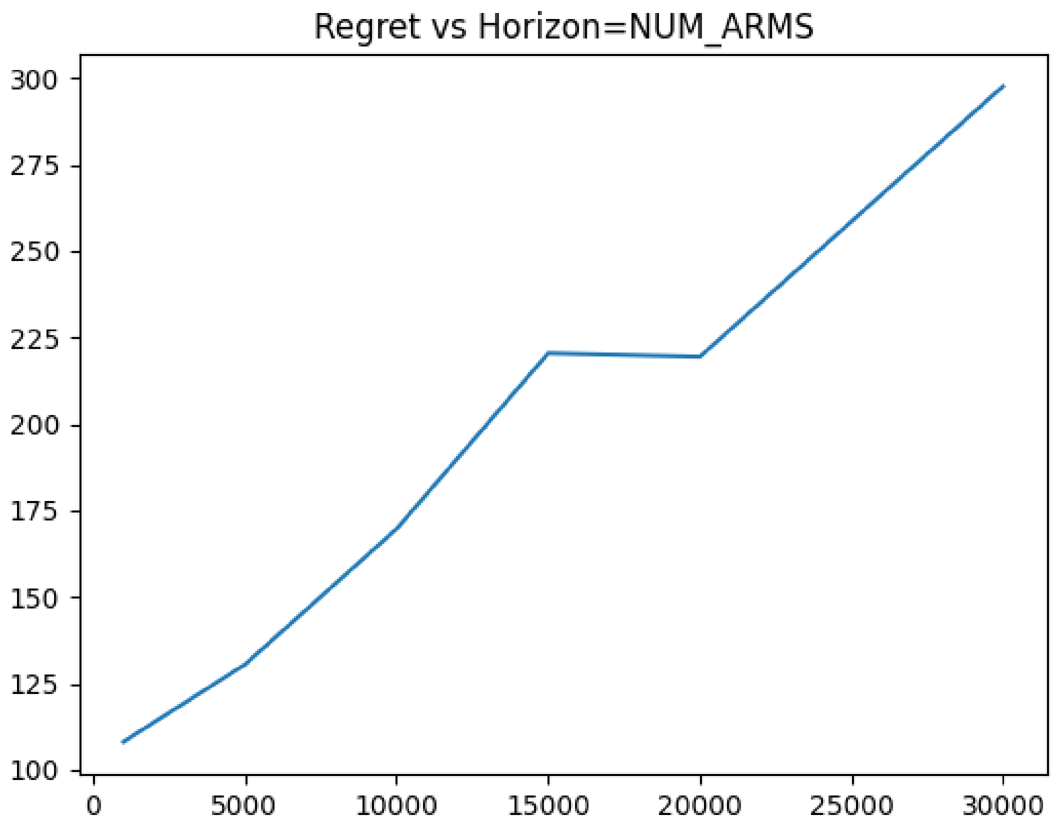
- The empirical values of the expected reward will be stored in an array and then the one with maximum value is being pulled for further exploitation



TASK 3:

The structure of the code is as :

- The first arm is pulled at random
- The succeeding arms are pulled depending on the reward obtained from the last pulled arm:
    1. IF REWARD FROM PREVIOUS ARM ==1: then the same arm is pulled again
    2. IF REWARD FROM PREVIOUS ARM == 0: then that means the empirical mean of reward from that arm would be definitely less than or equal to max of optimal mean reward: Hence we will check if the empirical mean reward uptill that instance is grater than 0.98 then we will pull the same arm otherwise we will randomly explore new arm from the rest set of arm(rest means: the arms which haven't been pulled even at least once uptill now)
    3. Here 0.98 is tuning parameter. By changing it we can get the regrets different for different cases.
- Following is the plot i am getting for task 3:


Regret vs Horizon=NUM_ARMS