Reinforcement Learning homework 1:

Algorithm:

The e-greedy algorithm was implemented in the following steps:

1) The action, reward and value matrices were initialized with 10x1000 zero matrices. The true value (a) for each action was sampled from a normal distribution, N(0,1). The reward for each action was randomly sampled from a normal distribution with its true value as the mean and the variance as 1, N(a,1).
2) For the first step, an action was chosen randomly, the reward was calculated and the action matrix was updated.
3) For the remaining steps, the value for each action was calculated using sample-average and the value matrix was updated.
4) At each step 't', the greedy option, that is, the action with the maximum value was chosen with a probability of 0.9 (1-e) and the non-greedy option, that is, an action was chosen at random with a probability of 0.1 (e). The action and reward matrices were updated.
5) The 1000 steps were run for 2000 iterations. The total reward for every step for every iteration was stored in a matrix (RG) and the average reward was calculated and plotted.

The implementation of the UCB algorithm was a little similar to the e-greedy algorithm:
For UCB, I used two value matrices. One for Sample-average and one for the sum of sample-average and the uncertainty term.

1) The action, reward and value matrices were initialized with 10x1000 zero matrices. The true value (a) for each action was sampled from a normal distribution, N(0,1). The reward for each action was randomly sampled from a normal distribution with its true value as the mean and the variance as 1, N(a,1).
2) For the first step, an action was chosen randomly, the reward was calculated and the action matrix was updated.
3) For the remaining steps, two values were calculated for each action, one as the sum of the sample-average and uncertainty term, and the other as the sample average. The Value matrices were updated.
4) At each step 't', the greedy option, that is, the action with the maximum value of sample average was chosen with a probability of 0.9 (1-e) and the non-greedy option, that is, the action with the maximum new value (sample average + uncertainty) was chosen with a probability of 0.1 (e). The action and reward matrices were updated.
5) The 1000 steps were run for 2000 iterations. The total reward for every step for every iteration was stored in a matrix (RGu) and the average reward was calculated and plotted.

Time elapsed: 666.92 seconds

Figures: