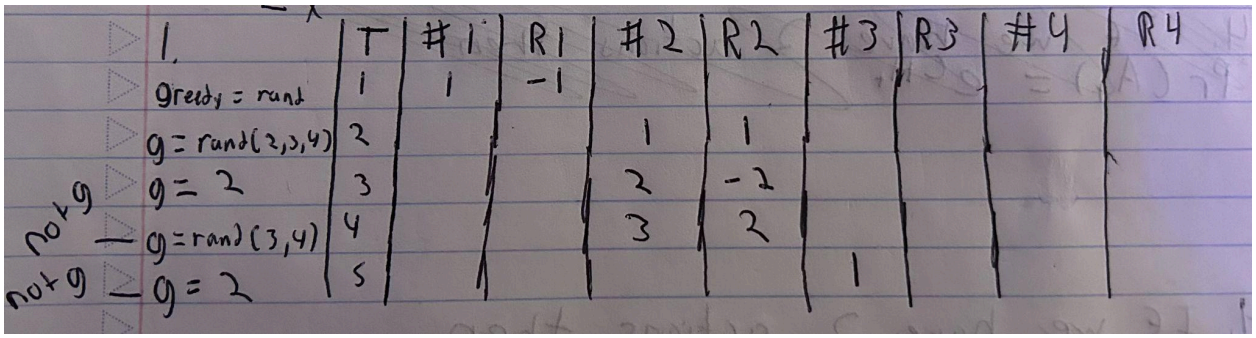
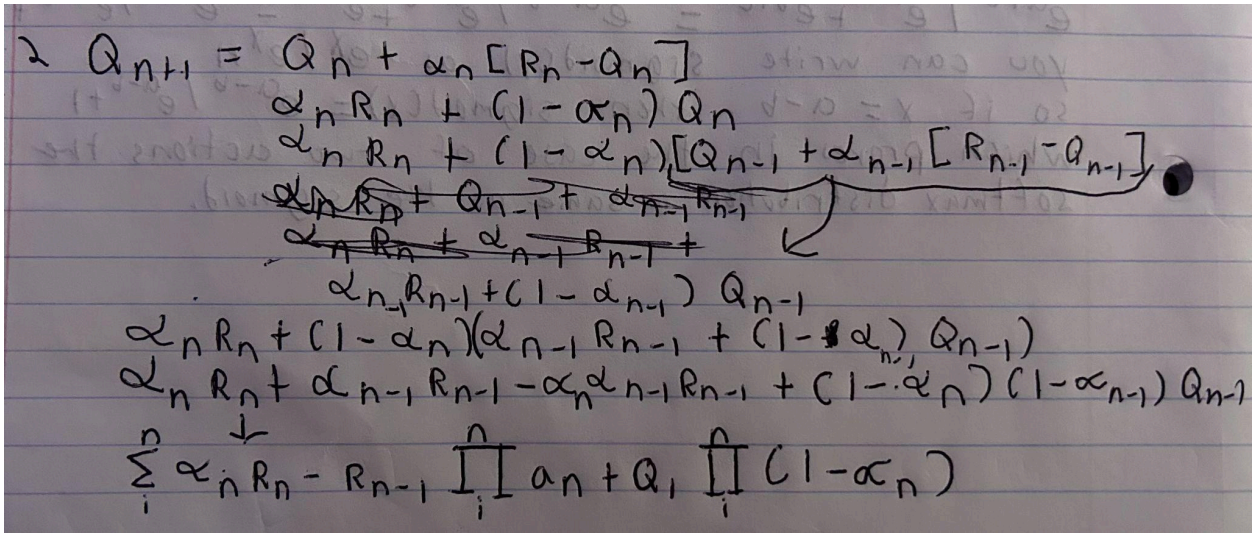
Miles Vollmer
RL Homework 2 EX-1
Q1:
Work:



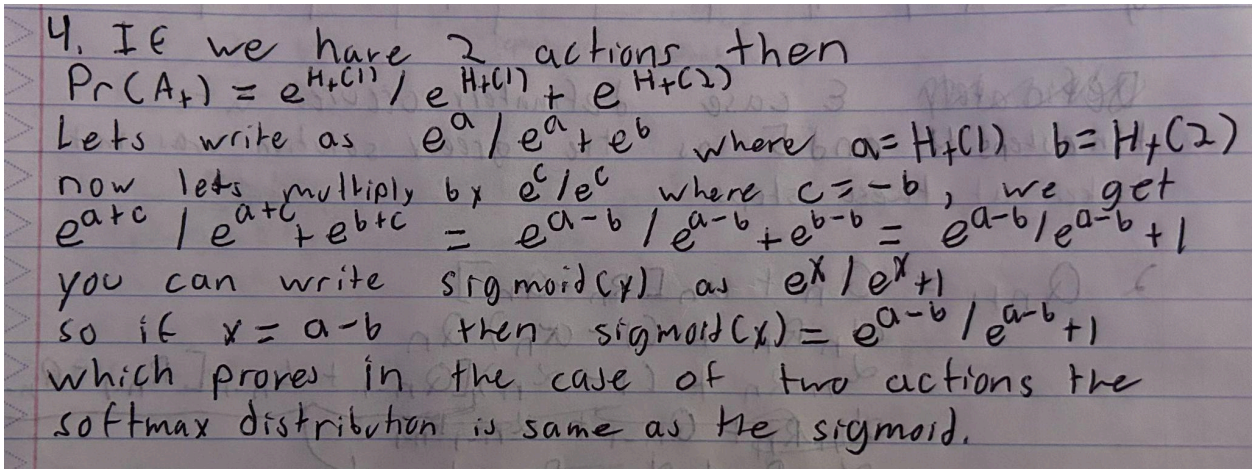
Explanation:
There was definitely an epsilon selection at timesteps 4 and 5. I know this because at timestep 1 both greedy and epsilon would just choose randomly, so can't guarantee an epsilon selection at this point. At timestep 2 greedy would choose randomly between actions 2, 3, and 4 and 2 is selected so can't guarantee an epsilon selection at this point. At timestep 3 greedy would choose 2 and 2 is selected so again can't guarantee an epsilon selection at this point. At timestep 4 greedy would randomly select between 3 and 4, as the expected reward for 1 and 2 are both negative at this point. However, 2 was selected at this point so we can guarantee that this was an epsilon selection. At timestep 5 greedy would select action 2 as its expected reward is now higher than any other action. However, 3 was selected at this point meaning that we can guarantee this was an epsilon selection.

Q2.
Work:



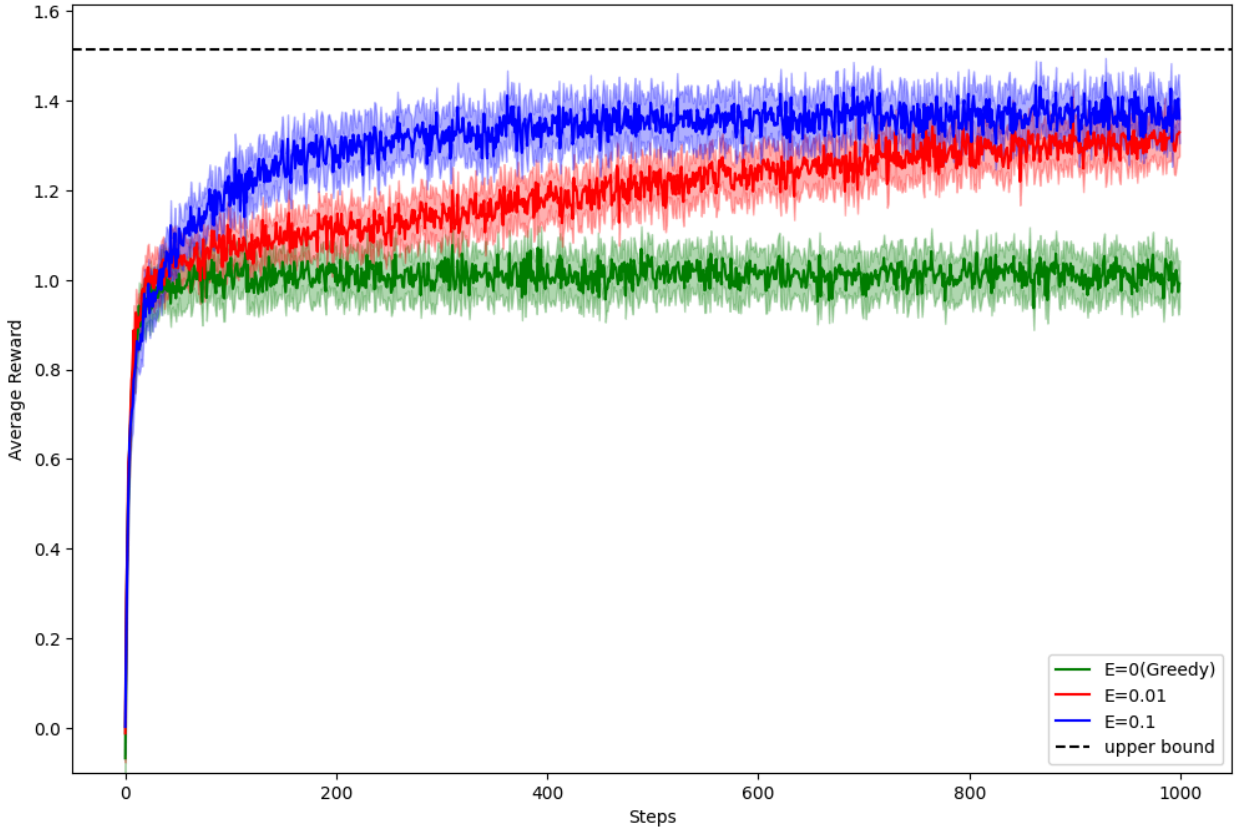
Explanation: If we take our equation 2.5 and break it down similarly to how it was done for 2.6 however in this case we don't consider alpha to be constant. This results in a weighting similar to that of 2.6 with some key differences(i.e. The new product portions)

Q4.
Work:

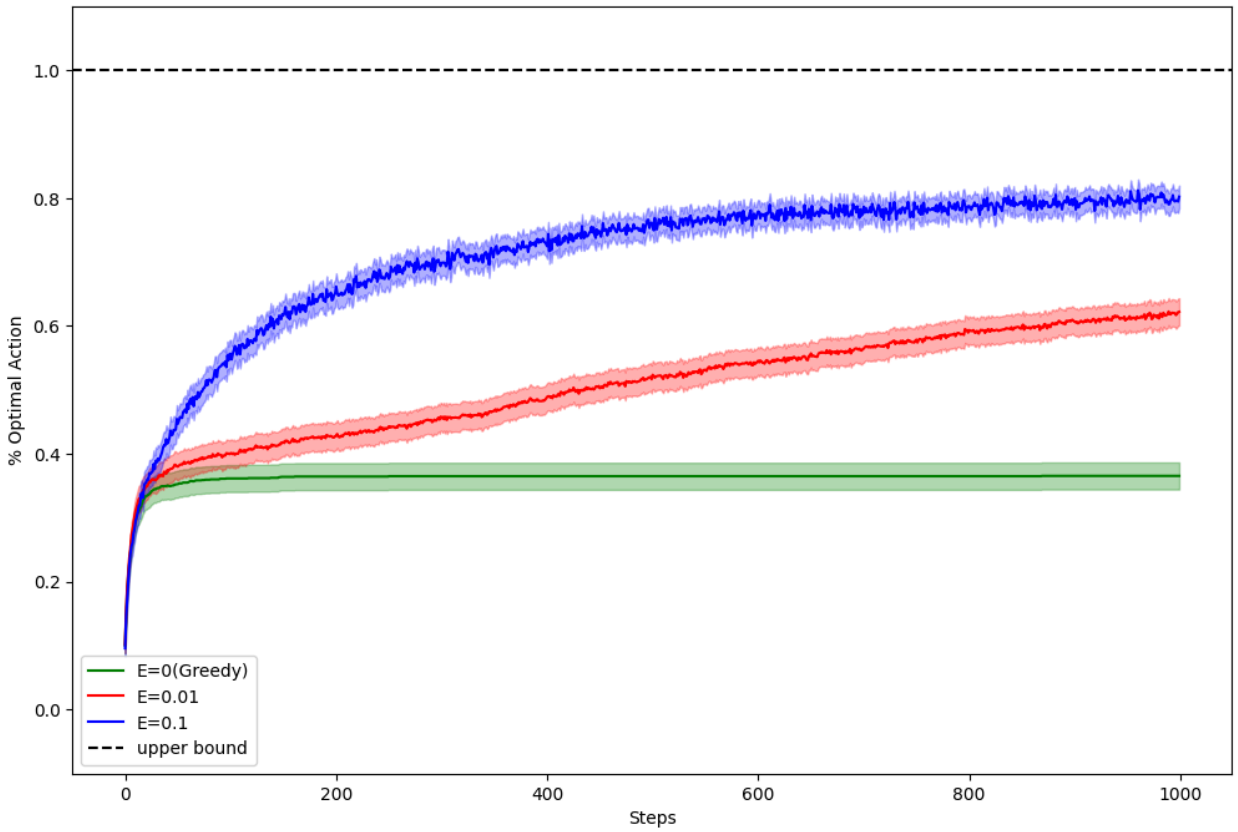


Explanation: see picture above

Q5.
Plot:



Average Reward of E-Greedy with different epsilon



Optimal Action Percentage of E-Greedy with different epsilon

Written: E=0 converges around 1, E=0.01 converges around 1.25, and E=0.1 converges around 1.35. The algorithms converge to different values due to their different epsilon values because these epsilon values dictate how much exploration the algorithm performs. So in this graph we see that the E=0.1 algorithm performs best over the 1000 steps. This is because the E=0 does no exploration and only chooses what it thinks is the best action, so it never learns the optimal and converges at 1. The E=0.01 algorithm is interesting because it performs worse over these 1000 time steps as it explores slower than E=0.1 as it is selecting what it thinks is the best option 99% of the time. However with E=0.01 once it learns the optimal action it picks it 9% more often than with E=0.1 so if we extended this graph we would see E=0.01 take over as the best performing algorithm.

Q6.
Plot:

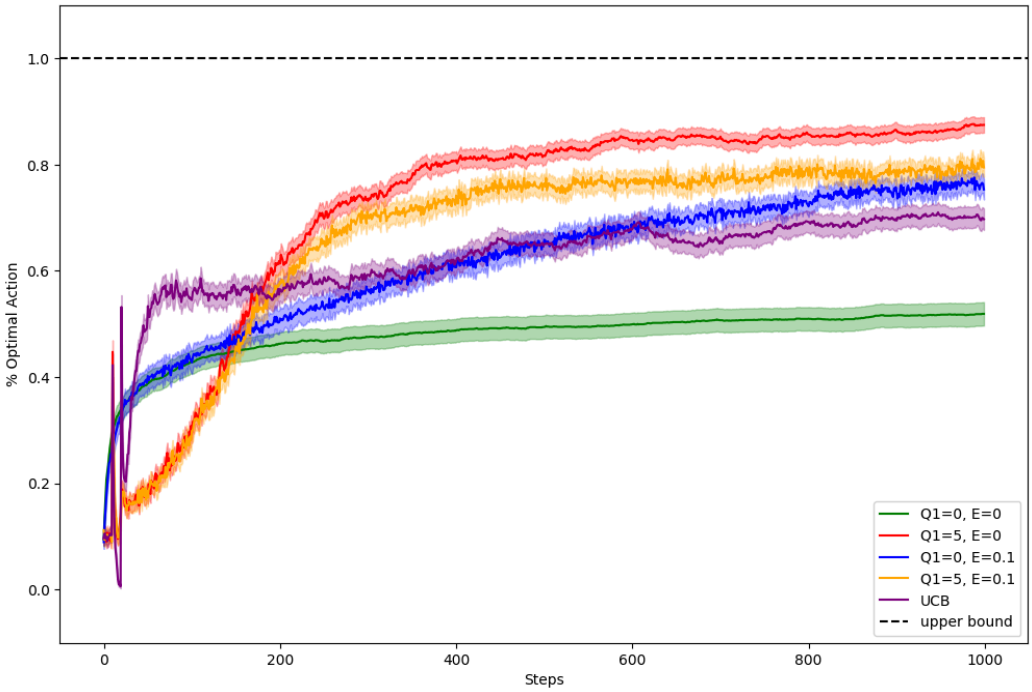


Fig 2.3: Reproduced

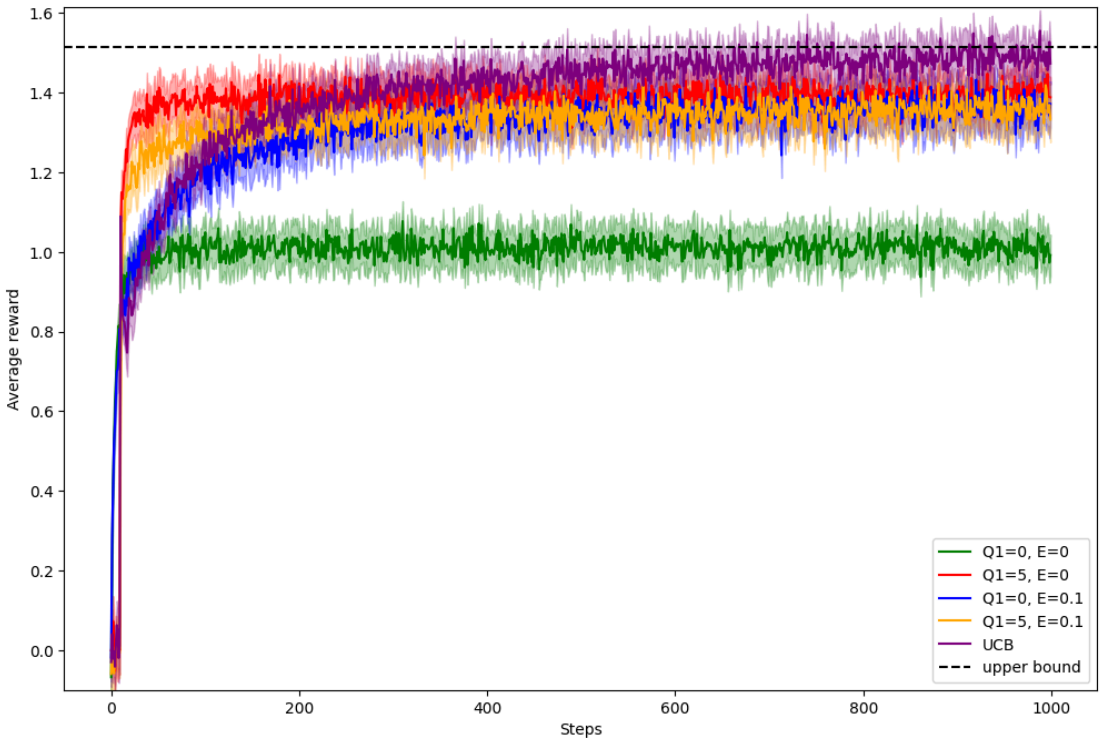


Fig 2.4: Reproduced

Written:
The spikes that appear in the UCB algorithms at the beginning of the two figures are due to how the UCB algorithm goes about its initial exploration. In the UCB algorithm, the initial surge in performance arises from the algorithm's strategy of favoring exploration, particularly for actions that have been chosen less frequently. By assigning higher values to these underexplored actions, UCB encourages the agent to investigate actions with greater uncertainty, characterized by larger confidence intervals. This exploration often results in a significant increase in cumulative rewards when these actions prove to be rewarding. However, as the agent gathers more data, the uncertainty about these actions decreases, and the confidence bounds shrink. Consequently, the agent shifts its focus away from exploration, leading to a decline in cumulative rewards when the agent learns that some actions were not as promising as initially expected. This process of early exploration followed by a correction phase accounts for the characteristic spikes in UCB's performance at the beginning of the learning process.