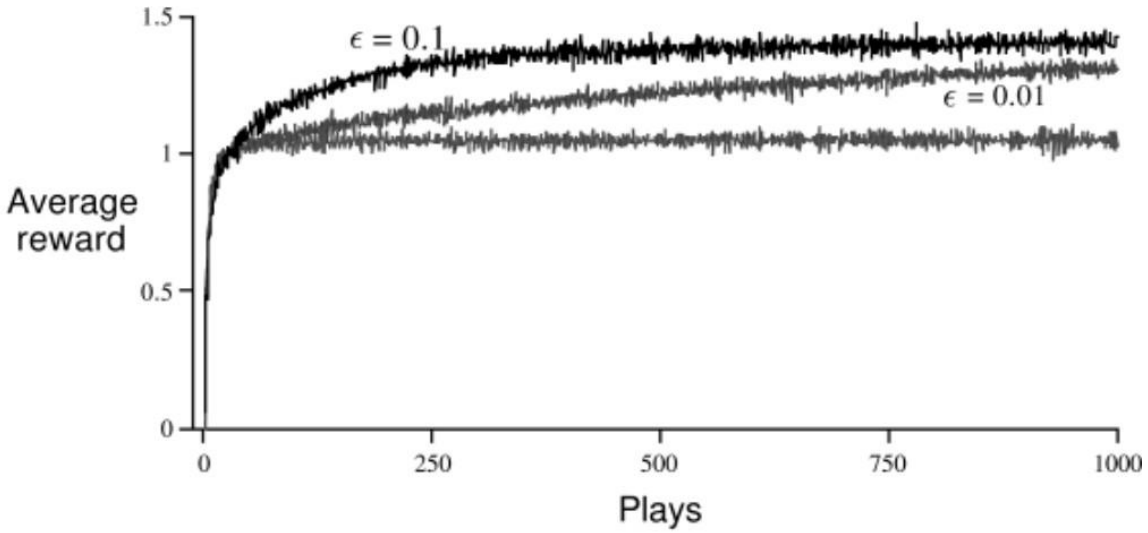
Homework 2: Multi-Armed Bandits

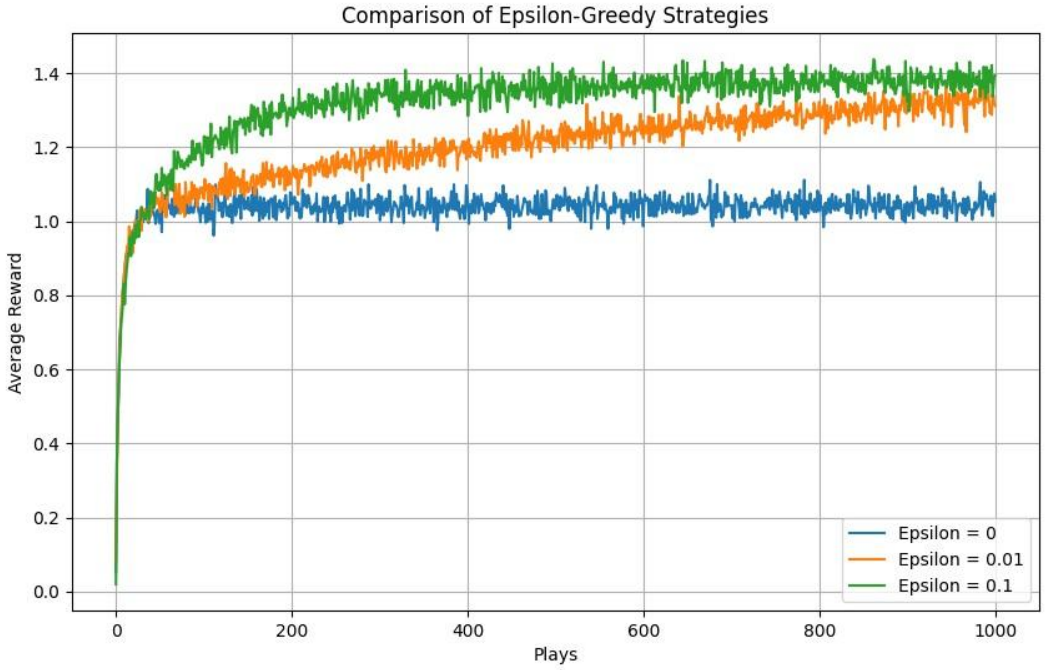
Atharv Kulkarni
U1322897

Problem 1: Your goal is to reproduce something similar to the top plot in Section 2.2, Figure 2.1 from Sutton and Barto. To make the experiment run faster, feel free to make the number of bandit tasks that you average over smaller than 2000 (but do at least 100). Discuss how your results compare with those in the Sutton and Barto book (http://incompleteideas.net/book/ebook/node16.html). Note you only have to plot the average reward, not the optimal action, but you should have lines for epsilon = 0, 0.01, and 0.1.



1. Discussion of the book figure:

- ε = 0 (Greedy Method):
  - Starts with a sharp rise but quickly plateaus at a lower average reward (~1).
  - Gets stuck in suboptimal actions because it doesn't explore after initial bad samples.
- ε = 0.01:
  - Slower improvement initially but eventually surpasses ε = 0.
  - Achieves a higher long-term reward due to occasional exploration, discovering better actions over time.
- ε = 0.1:
  - Fastest growth in average reward due to more frequent exploration.
  - Approaches a higher average reward (~1.4) than both ε = 0 and 0.01.



2. Discussion of the my results:

- ε = 0 (Greedy Method):
  - Matches the book's behavior: initial improvement, then flatlining at a lower reward.
  - This is due to the same problem—getting stuck in suboptimal arms.
- ε = 0.01:
  - Similar to the book: slow improvement, eventually outperforming the greedy method.
  - This shows the effect of occasional exploration, though not as aggressive as ε = 0.1.
- ε = 0.1:
  - Shows rapid growth and achieves the highest average reward, similar to the book's plot.

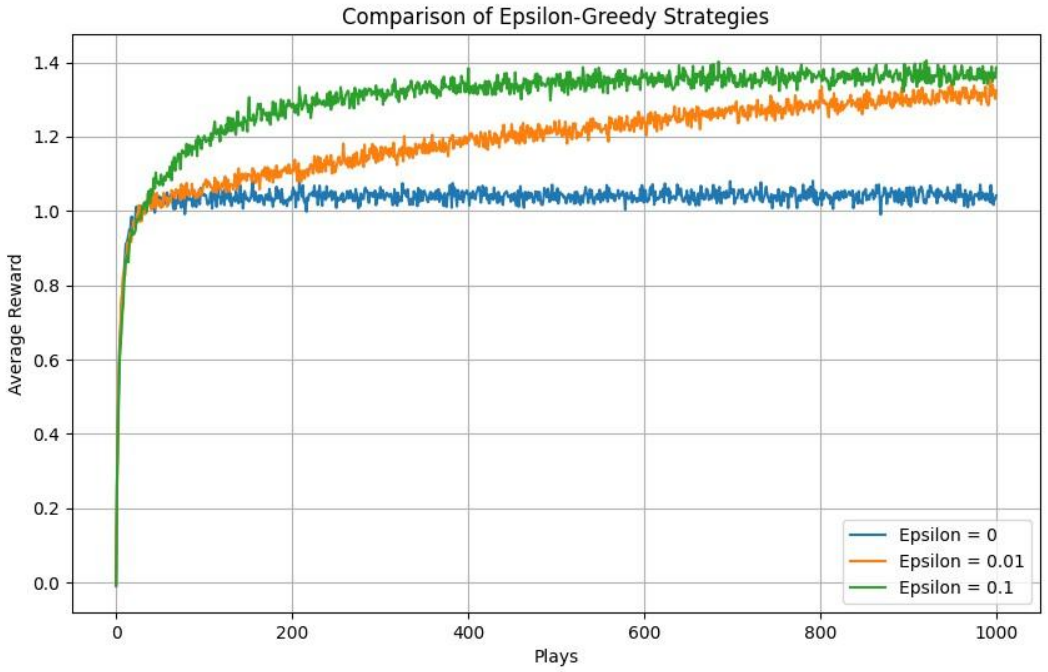
3. Assumptions I made in my implementation:

1. Gaussian Reward Distribution:
   - Rewards are sampled from N(0, 1), matching the book's description. ○ This ensures that the reward variance aligns with expectations.
2. Sample-Average Method:
   - Uses the incremental update rule for efficiency (as discussed in Section 2.5 of the book).

○ This avoids the need to store all past rewards, which is consistent with Sutton & Barto's approach.

3. Averaging Over Multiple Bandit Tasks:
   ○ Averaging over 2000 tasks (or the n_tasks), as recommended in the book. ○ This helps reduce variance and provides smoother plots.
4. Epsilon-Greedy Policy:
   ○ The exploration rate (ϵ) is fixed during the experiment.
   ○ Matches the book's static exploration approach.

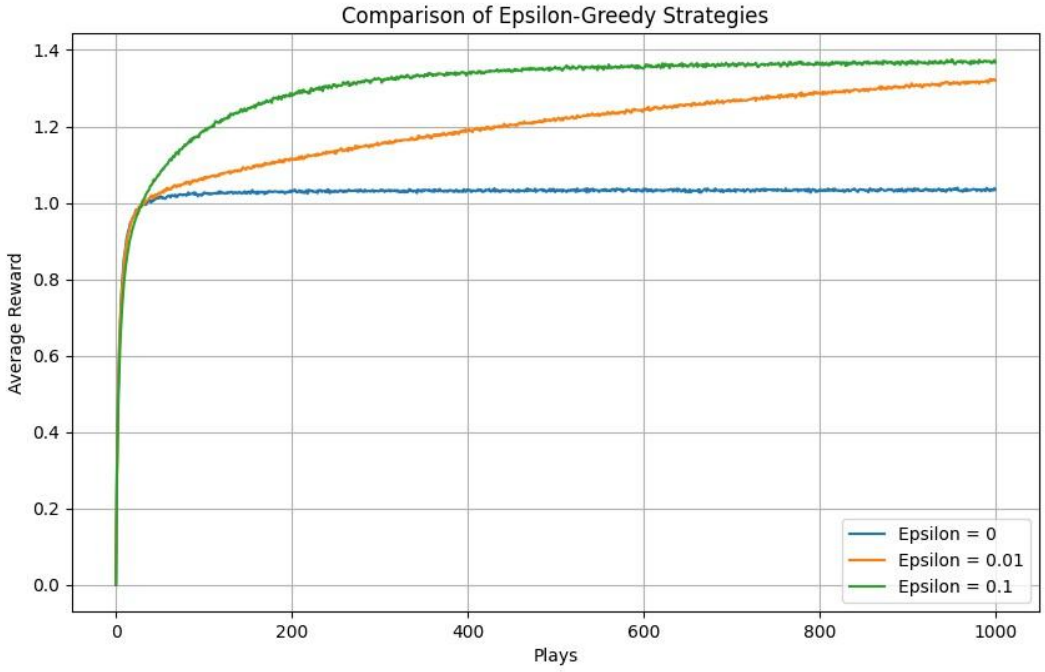
N_tasks = 5000:



Observations for 5000 Tasks:

- Smoother Curves:
  - Compared to 2000 tasks, the plot is significantly less noisy, indicating better averaging.
  - The gap between ϵ = 0.1 and ϵ = 0.01 is clear, with ϵ = 0.1 consistently outperforming.
- The overall trends are still consistent to when the tasks were 2000.

N_tasks = 200000:



Observations for 200000 Tasks:

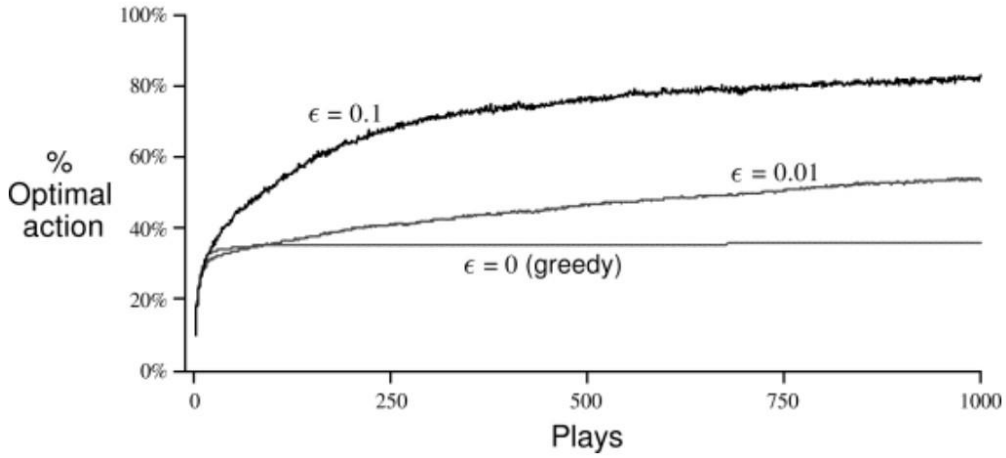
- Smoother Curves:
  - Compared to 2000, 5000 tasks, the plot has very little noise, indicating better averaging.
  - The gap between ϵ = 0.1 and ϵ = 0.01 is clearer, with ϵ = 0.1 consistently outperforming.
- The overall trends are still consistent to when the tasks were 2000.

Conclusion:

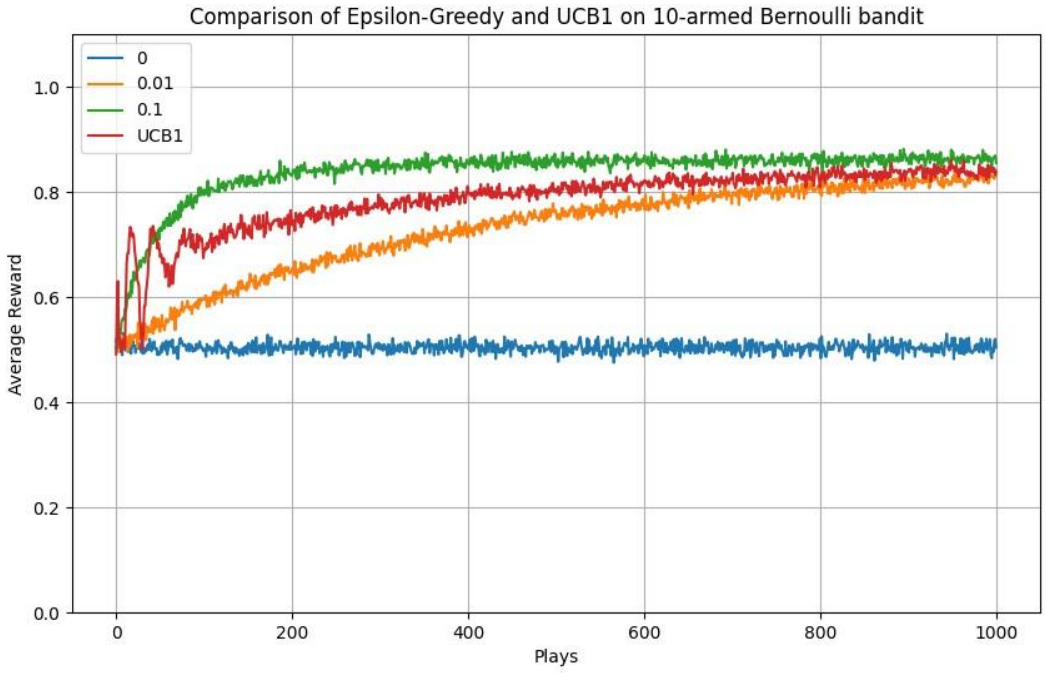
- ϵ = 0.1 balances exploration and exploitation best, finding optimal actions faster.
- ϵ = 0.01 explores less aggressively, improving slowly but steadily. ● ϵ = 0 (greedy) stagnates early due to no exploration.

Problem 2: For this question, we will consider the setting of a Bernoulli bandit. This is the simplest bandit setting where each arm outputs a reward of +1 with probability p and a reward of 0 with probability 1-p. Run an experiment where you have a 10-armed Bernoulli bandit problem and compare epsilon-greedy epsilon = 0, 0.01, and 0.1 with UCB1. Because results will

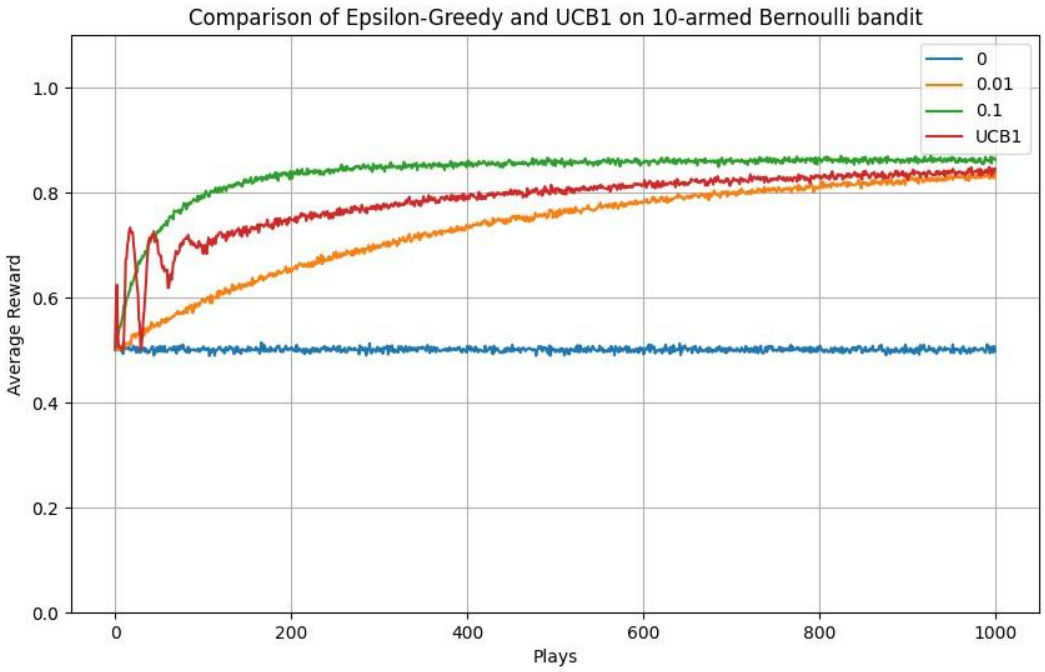
be noisy, average your results over 100 random bandit problems. To create each problem you should sample a probability p for each arm uniformly from the range[0,1]. Discuss and interpret your results.



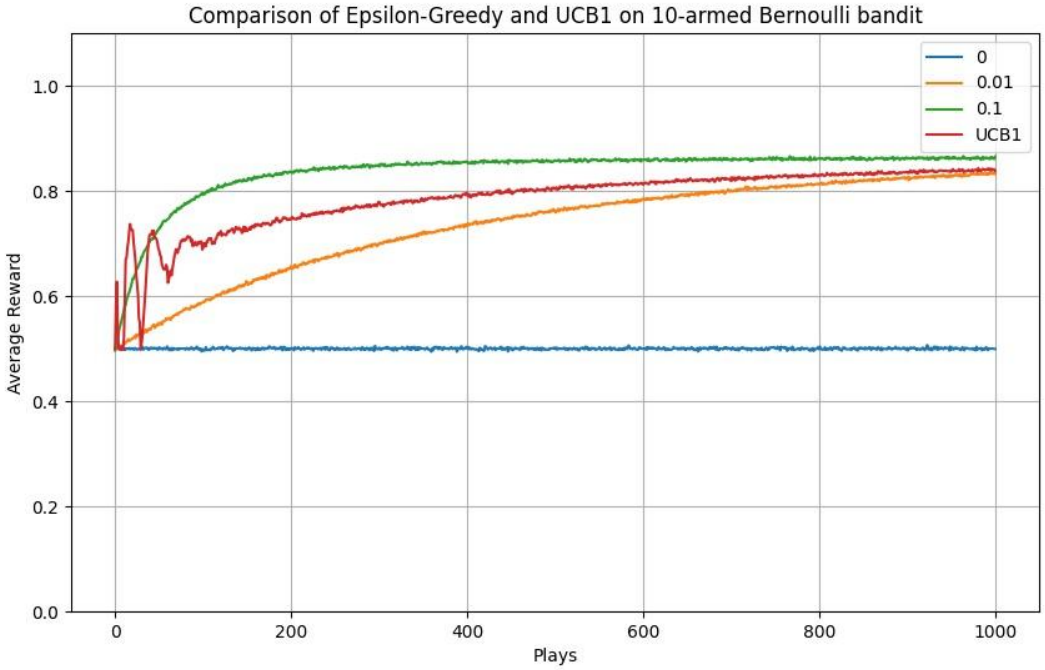
N_Tasks = 2000:



N_Tasks = 10000:



N_Tasks = 50000:



UCB1 Algorithm
(https://www.geeksforgeeks.org/upper-confidence-bound-algorithm-in-reinforcement-learning/)**:**

I implemented the UCB1 algorithm based on the above link. This is designed to balance exploration and exploitation in an intelligent way. It selects actions based on two factors:

1. **Action Estimates:** The average reward received from each arm.
2. **Confidence Bound:** A term that encourages exploration, calculated as $\sqrt{(\ln(t) / n_i)}$, where:
    - t is the total number of steps taken so far.
    - n_i is the number of times arm i has been selected.

At each step, UCB1 picks the arm with the highest sum of the action estimate and the confidence bound. This ensures that arms with uncertain rewards (those that haven't been tried much) still have a chance to be selected, while arms with consistently good rewards are exploited.

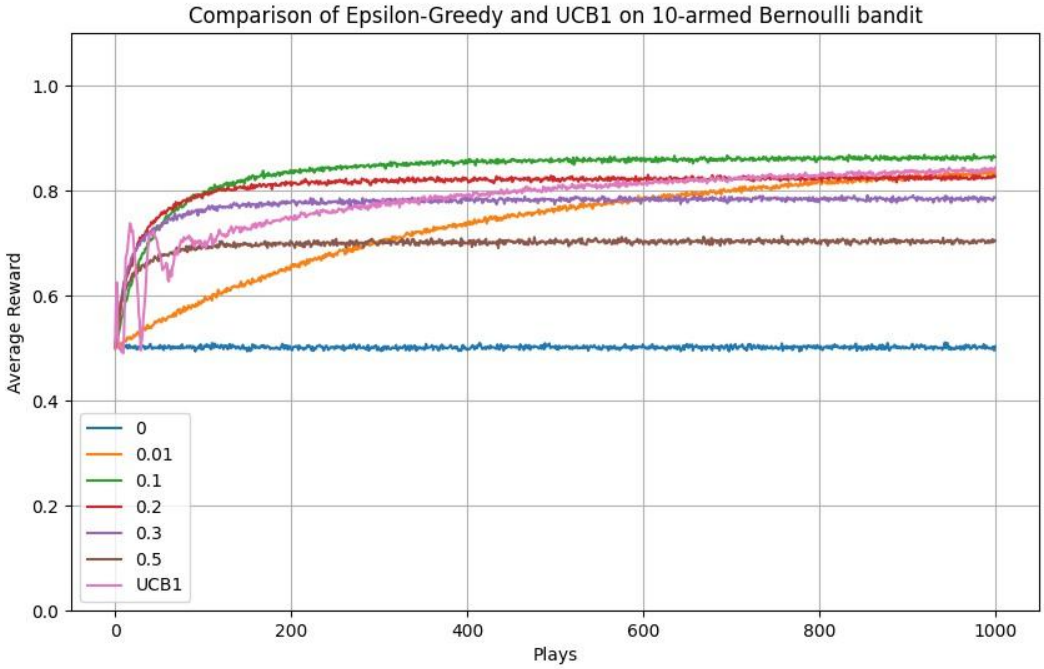
Key Observations:

- **UCB1 Performance:** UCB1 performed competitively with epsilon-greedy strategies, but it did not consistently outperform all epsilon-greedy strategies in all cases. While UCB1 demonstrated strong performance in certain scenarios, particularly in the early stages, epsilon-greedy with $\varepsilon = 0.1$ showed slightly better long-term performance as the number of plays increased. The gap between UCB1 and epsilon-greedy strategies became narrower as we increased the number of tasks from 2000 to 10000 and then to 50000. However, across other epsilon values, UCB1 performed better and only fell behind $\varepsilon = 0.1$ (See plot below).
- **Epsilon-Greedy Strategies:** Among epsilon-greedy methods, $\varepsilon = 0.1$ did better than $\varepsilon = 0.01$ and $\varepsilon = 0$. A higher ε means more exploration, which helps find better arms as discussed in problem 1.
- **Pure Exploitation ($\varepsilon = 0$) Behavior:** The $\varepsilon = 0$ strategy (pure exploitation) performed the worst because it doesn't explore, getting stuck with arms that seemed good early on but aren't the best. This effect was even more noticeable in larger task sets like 50000 tasks.

Comparison with Sutton and Barto:

The results match what Sutton and Barto found. We saw the same trend where $\varepsilon = 0.1$ worked better than $\varepsilon = 0.01$ and $\varepsilon = 0$. UCB1's performance supports their ideas about the importance of balancing exploration and exploitation. As the number of tasks increased, the consistency of these trends further confirmed the reliability of these strategies.

UCB1 Performance across a variety of epsilons:


Comparison of Epsilon-Greedy and UCB1 on 10-armed Bernoulli bandit
Average Reward
Plays
0.0
0.2
0.4
0.6
0.8
1.0
0
200
400
600
800
1000
0
0.01
0.1
0.2
0.3
0.5
UCB1