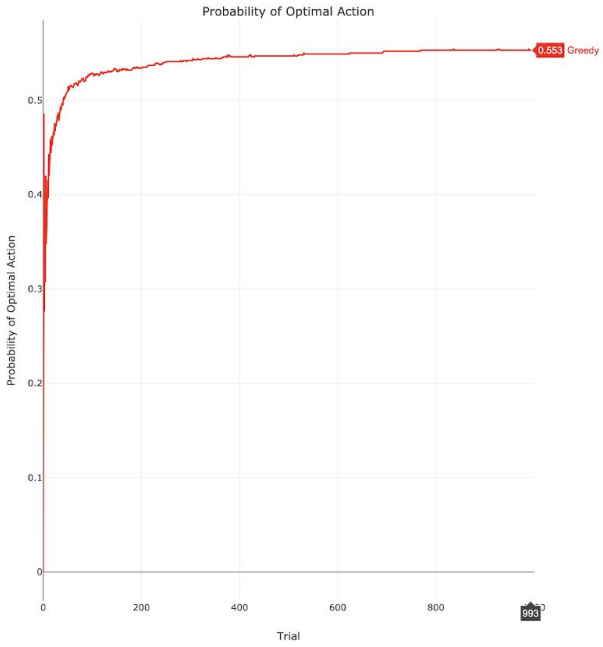
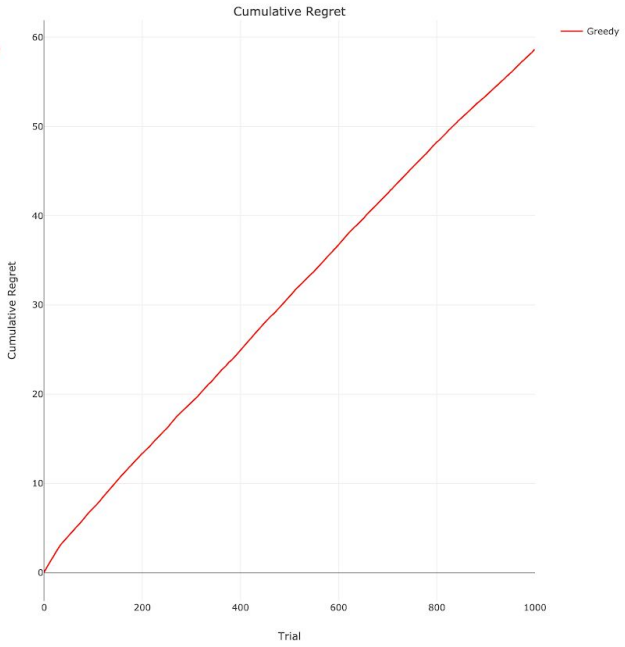
Problem 1: Greedy Agent

Expected:
The Greedy agent is expected to find a good solution that returns correct over 50% of the time, but not always the optimal. Cumulative regret is expected to be stable at a linear rate.

Sim Results



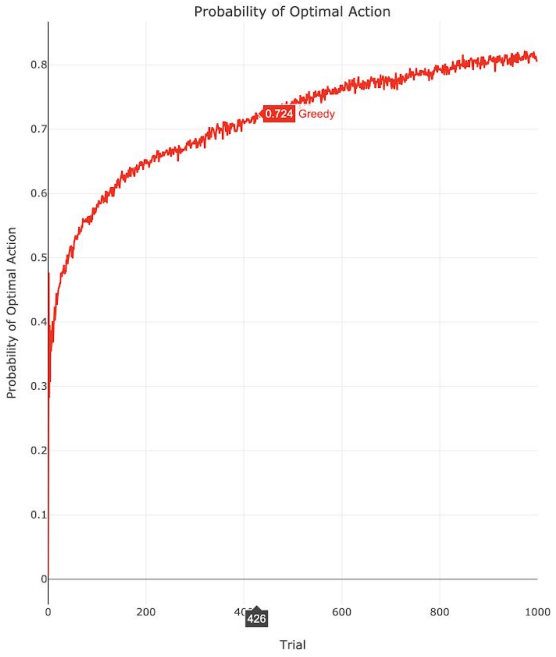


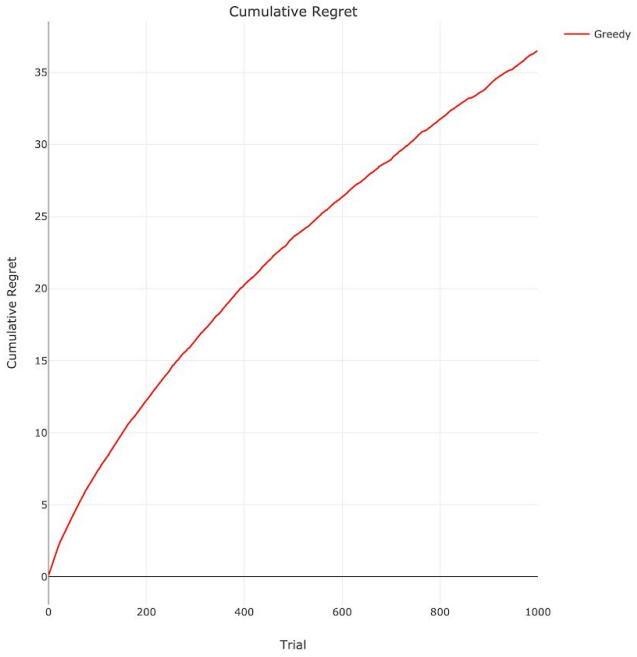
Problem 2

Expected
Epsilon Greedy is expected to converge the probability of finding the optimal answer as 1 - epsilon. Cumulative regret should be logarithmic as time goes on.

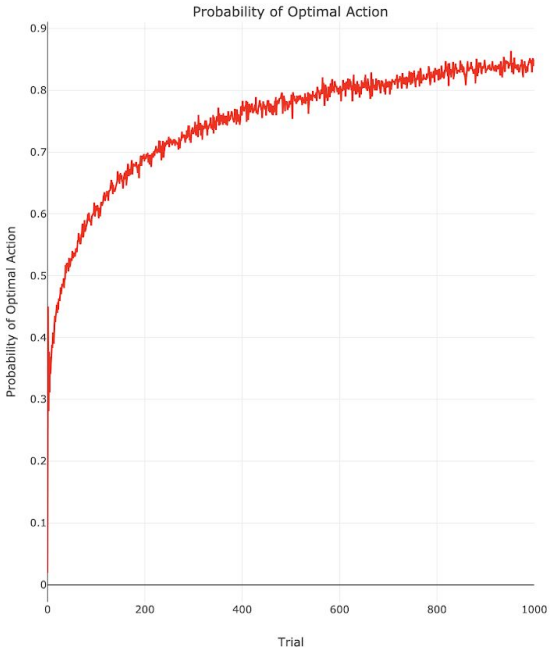
Sim Results

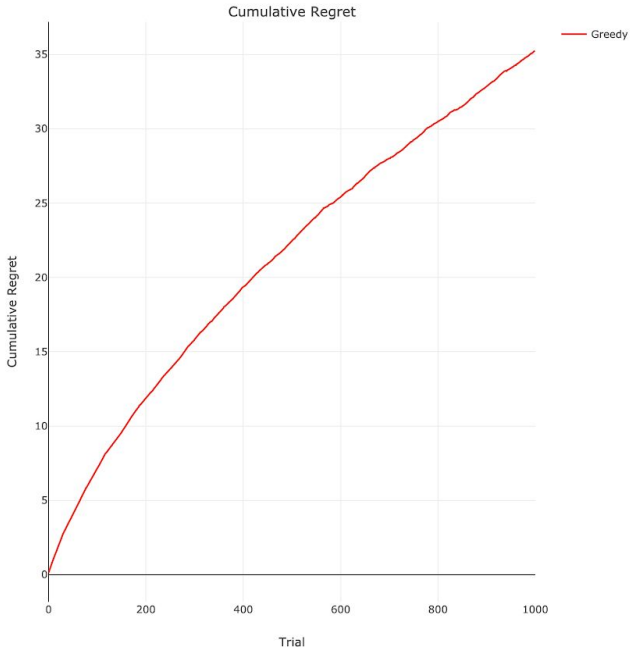
E=0.05



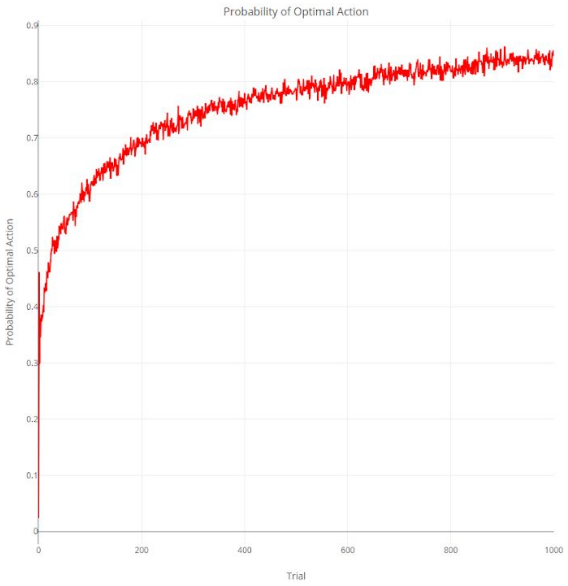


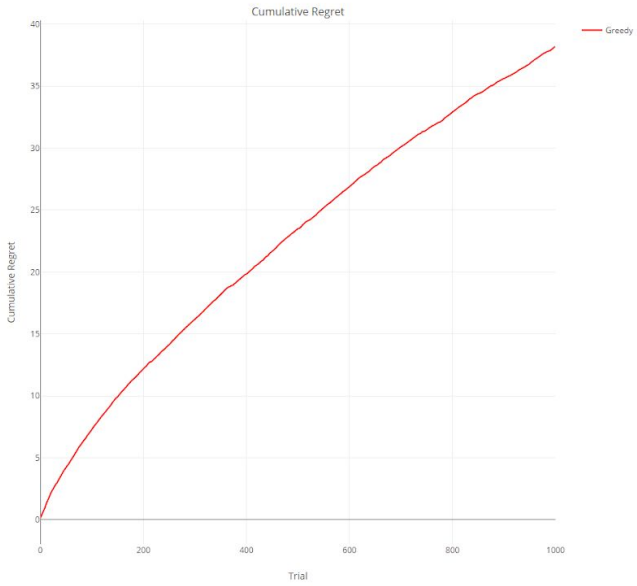
E=0.10:


Probability of Optimal Action
Probability of Optimal Action
Trial
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
200
400
600
800
1000


Cumulative Regret
Greedy
Cumulative Regret
Trial
0
5
10
15
20
25
30
35
0
200
400
600
800
1000

E = 0.15


Probability of Optimal Action
Probability of Optimal Action
Trial
0
200
400
600
800
1000


Cumulative Regret
Greedy
Cumulative Regret
Trial
0
5
10
15
20
25
30
35
40
0
200
400
600
800
1000

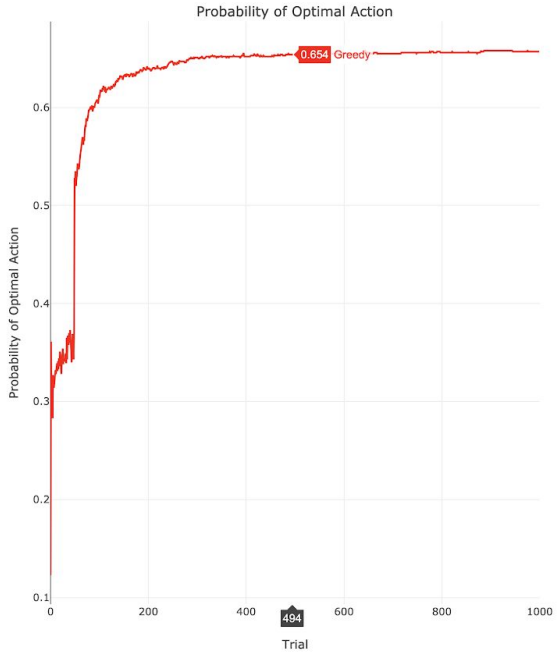
Problem 3

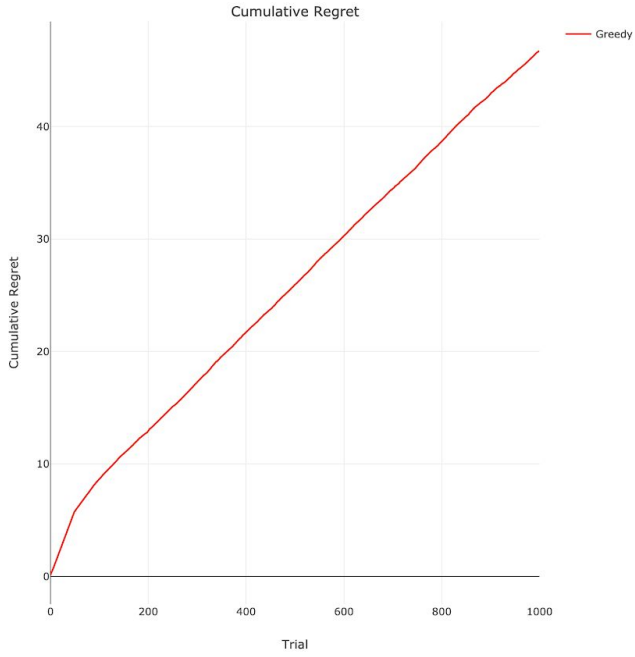
Expected

Knowing we have a finite time horizon T, we determined our values of epsilon based off of the results we got from the epsilon greedy approach. The values of epsilon should be fairly similar for epsilon first and for epsilon greedy.

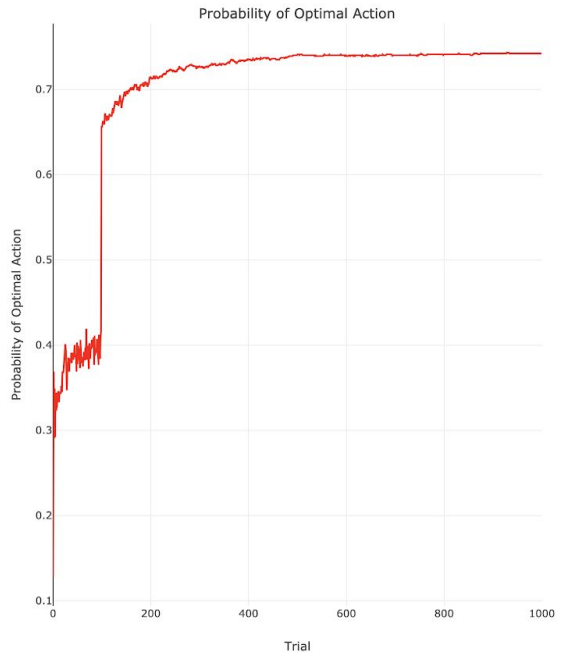
Sim Results

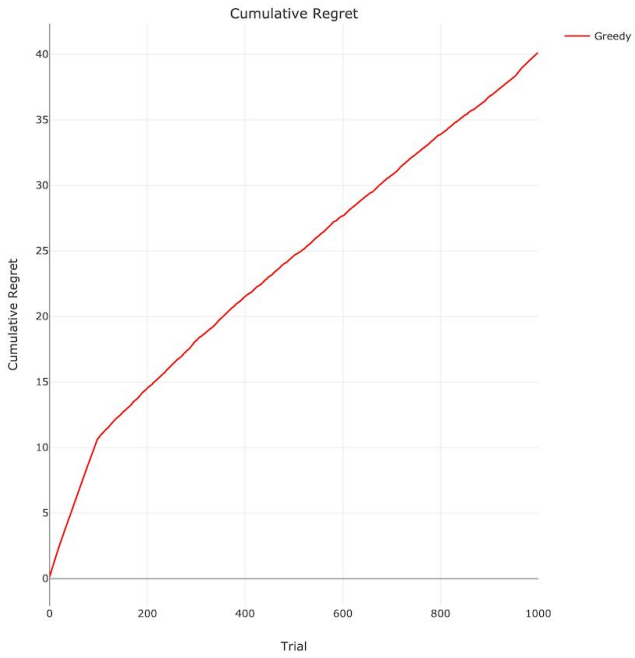
E=0.05



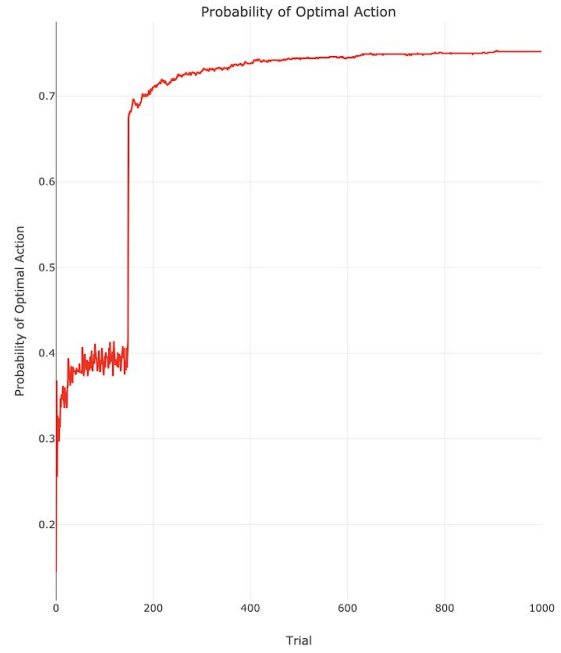


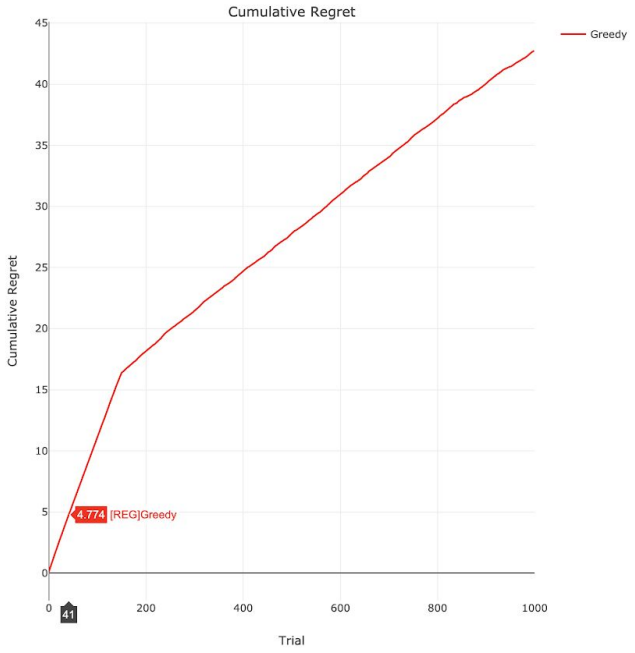
E=0.10


Probability of Optimal Action
Probability of Optimal Action
Trial
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
200
400
600
800
1000


Cumulative Regret
Greedy
Cumulative Regret
Trial
0
5
10
15
20
25
30
35
40
0
200
400
600
800
1000

E=0.15


Probability of Optimal Action
Probability of Optimal Action
Trial
0.2
0.3
0.4
0.5
0.6
0.7
0
200
400
600
800
1000


Cumulative Regret
Greedy
4.774 [REG]Greedy
41
Cumulative Regret
Trial
0
5
10
15
20
25
30
35
40
45
0
200
400
600
800
1000

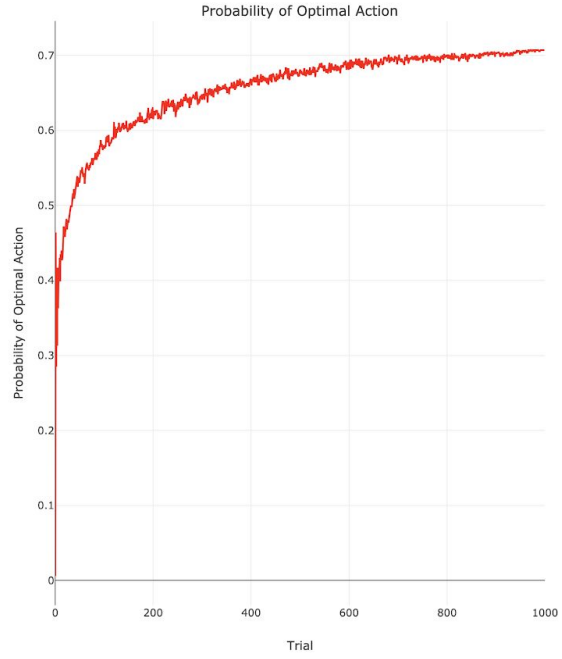
Problem 4

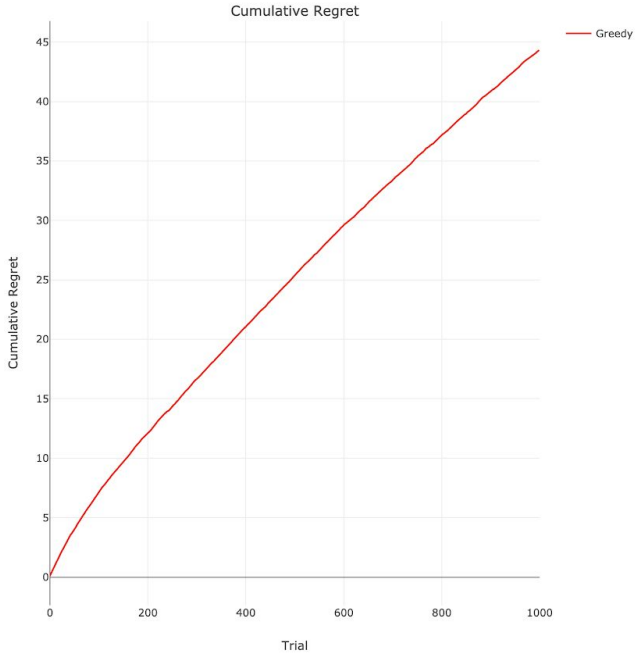
Expected

Sim Results

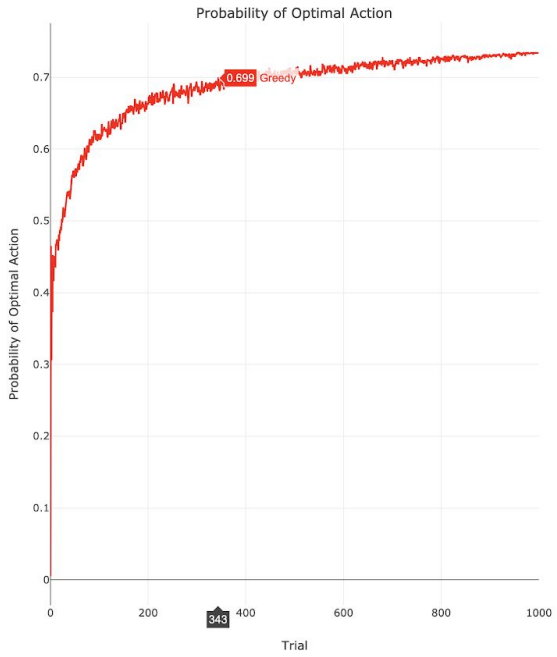
linear decrease: (e*(1-(current trial/total trials)))

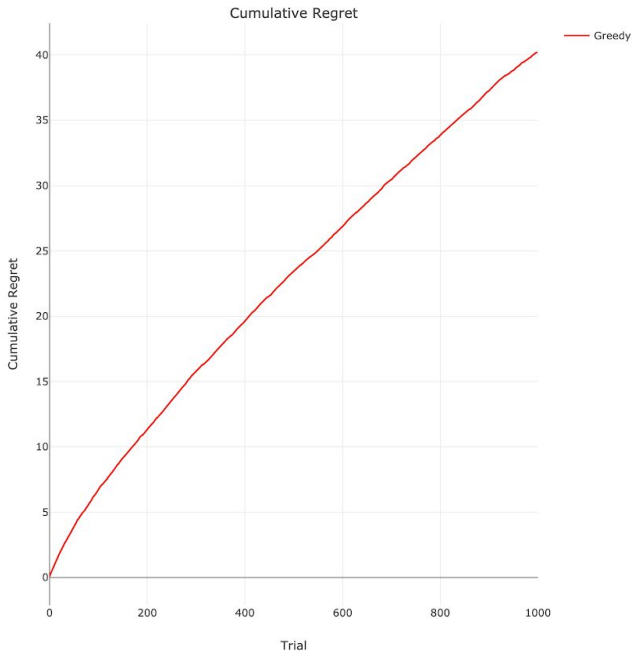
E = 0.05



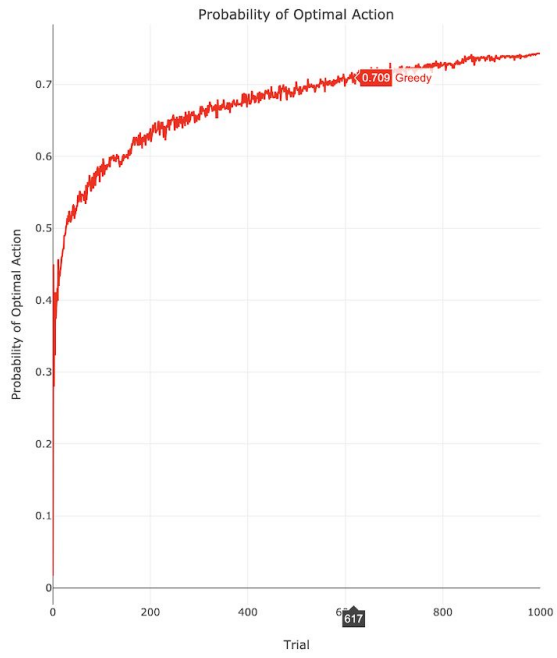


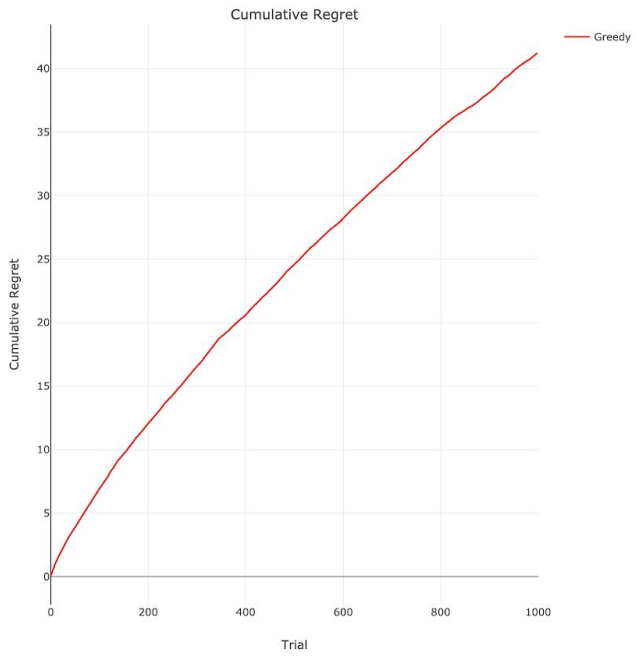
E = 0.10


Probability of Optimal Action
Probability of Optimal Action
0.699
Greedy
343
Trial


Cumulative Regret
Cumulative Regret
Greedy
Trial

E = 0.15

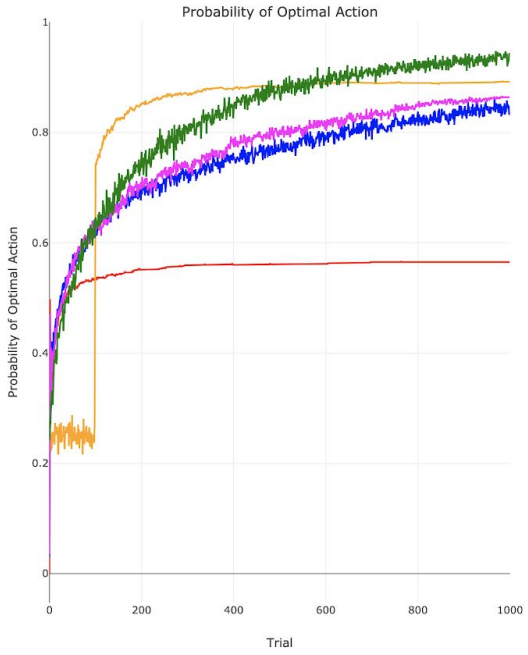

Probability of Optimal Action
Probability of Optimal Action
0.709
Greedy
617
Trial

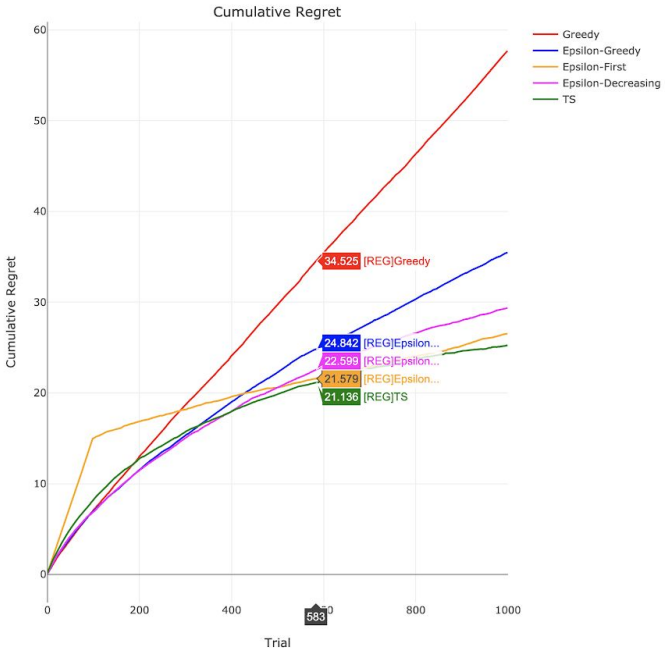

Cumulative Regret
Cumulative Regret
Greedy
Trial

Problem 5

Expectations
We used the np.random.beta function to run Thompson Sampling

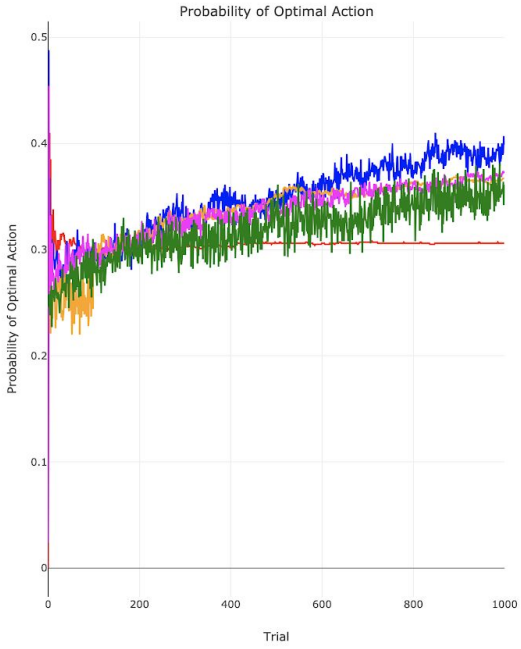
Sim Results

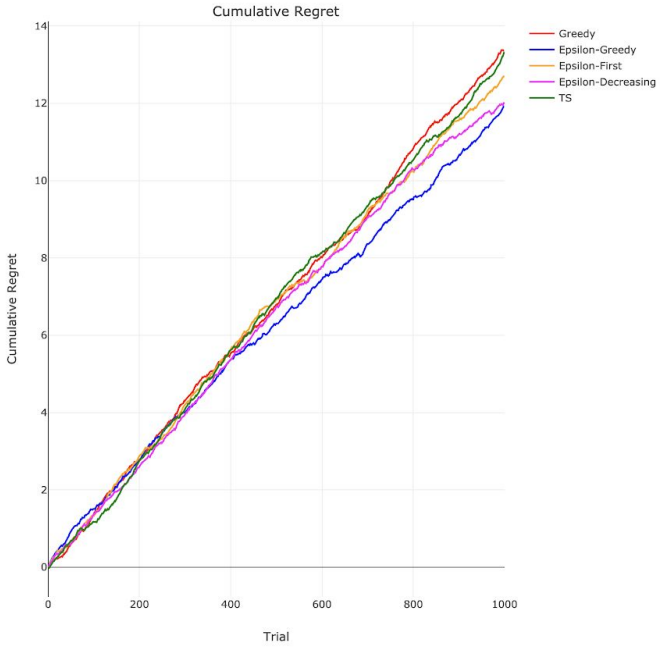




Compare

```
# Reward Signal Probabilities
# P(R_t = 1 | do(A_t))
P_R = np.array(
    # A =  0     1     2     3
    [0.49, 0.52, 0.51, 0.50]
)
K = len(P_R)
```





Reflect

All of the approaches are much closer to each other having the rewards be so close. Having the reward signals so close naturally decreases overall regret. Having them so close, though, makes it harder to truly identify the best choice for the problem. If we did not know or did not have a finite time T, Thompson Sampling would end up having the least cumulative regret. As it continues, it will do a better job identifying the optimal choice compared to the other options.