A)

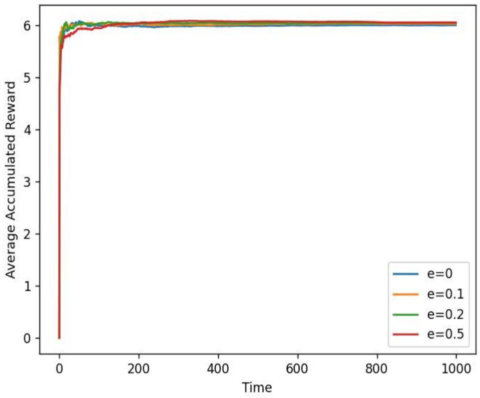


Figure 1 Average Accumulated Reward for α=1

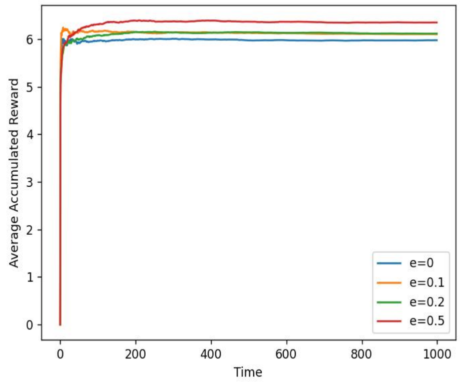


Figure 2 Average Accumulated Reward for $\alpha=0.9^k$

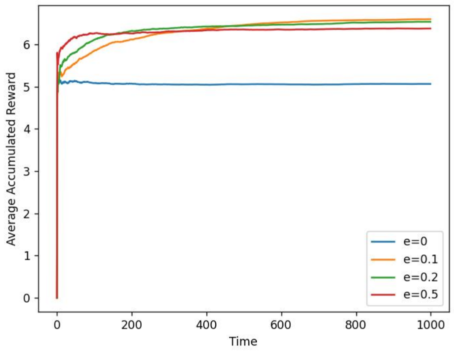
Table 1 Average Q-values α=1

Epsilon-greedy	Average of action value Q(a1) of 100 runs	**True action value $Q^*(a^1)$**	Average of action value Q(a2) of 100 runs	**True action value $Q^*(a^2)$**
ε = 0 (greedy)	5.00920326	**5**	6.94091167	**7**
ε = 0.1	4.84934599	**5**	6.85077403	**7**
ε = 0.2	4.94948212	**5**	6.61051014	**7**
ε = 0.5(random)	4.76514626	**5**	6.34843023	**7**

Table 2 Average Q-values $\alpha=0.9^k$

Epsilon-greedy	Average of action value Q(a1) of 100 runs	**True action value $Q^*(a^1)$**	Average of action value Q(a2) of 100 runs	**True action value $Q^*(a^2)$**
ε = 0 (greedy)	4.92406251	**5**	6.9617018	**7**
ε = 0.1	5.04772503	**5**	6.82325321	**7**
ε = 0.2	4.85349616	**5**	6.63852045	**7**
ε = 0.5(random)	4.47139182	**5**	6.49080936	**7**

According to Plot1, the ε = 0.1 perform the best, the graph depicts a rise in the anticipated reward with accumulated experience. In the early stages, the greedy strategy demonstrated a slightly faster improvement compared to the other methods, but then stabilized at a lower plateau. But it changed when change the leaning rate to 0.9k, in the plot 2, the policy ε = 0.5(random) seems get the best average accumulated reward. However the plot2 may don't correct is that when we execute the e-greedy policy, with less value of e show you can exploitation more, but the e shouldn't be too small that is because the exploration will be slowly.So maybe the best learning in part2 is ε = 0.1.



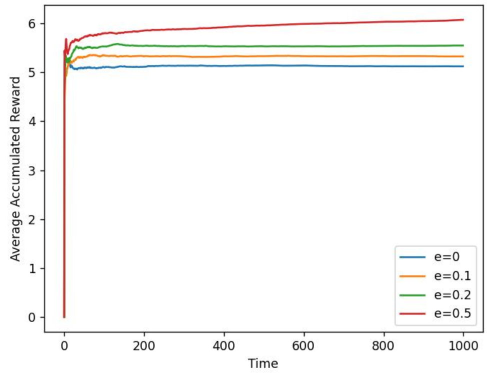


Figure 3 Average Accumulated Reward for α=1/(1+ln(1+k)) *Figure 4 Average Accumulated Reward for α=1/k*

Table 3 Average Q-values α=1/(1+ln(1+k))

Epsilon-greedy	Average of action value Q(a1) of 100 runs	**True action value $Q^*(a^1)$**	Average of action value Q(a2) of 100 runs	**True action value $Q^*(a^2)$**
ε = 0 (greedy)	5.00144144	**5**	6.7187843	**7**
ε = 0.1	4.98310878	**5**	6.71416822	**7**
ε = 0.2	4.94530912	**5**	6.68396624	**7**
ε = 0.5(random)	4.81967443	**5**	6.22457709	**7**

Table 4 Average Q-values α=1/k

Epsilon-greedy	Average of action value Q(a1) of 100 runs	**True action value $Q^*(a^1)$**	Average of action value Q(a2) of 100 runs	**True action value $Q^*(a^2)$**
ε = 0 (greedy)	4.99524356	**5**	6.69535422	**7**
ε = 0.1	4.94279144	**5**	6.70688896	**7**
ε = 0.2	4.96731042	**5**	6.67594623	**7**
ε = 0.5(random)	4.87941368	**5**	6.00660903	**7**

According to plot3, the ε = 0.1 show the best Average Accumulated Reward, but in the lopt4, the ε = 0.5(random) show the best Average Accumulated Reward. The plot 3 demonstrate that the greedy , ε = 0.1, and ε = 0.2 are all seems good to exploitation, they all rise so fast in the start and end with similar average accumulated reward for.

So, I would say the ε = 0.1, and learning rate = 1/(ln(k)+1) will maximum average accumulated reward.

B)

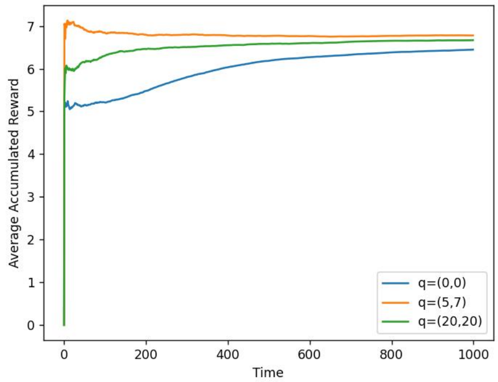


Figure 5. Average Accumulated Reward

Table5 Average Q-values with different initial Q

Initial Q values	Average of action value Q(a)1 of 100 runs	**True action value Q*(a1)**	Average of action value Q(a)2 of 100 runs	**True action value Q*(a2)**
Q =[0 0]	4.67315095	**5**	6.64070311	**7**
Q =[5 7]	4.53589951	**5**	7.00368398	**7**
Q =[20 20]	4.60372112	**5**	6.81512228	**7**

If the initial values are too high like Q=(20,20), which may overestimate the value of certain actions and choose them more often, leading to suboptimal results, and if the initial values are too low like Q=(0,0), which may underestimate the value of certain actions and not choose them enough.

So as far as I'm concerned, the optimal value of initial Q=(5,7).

C)

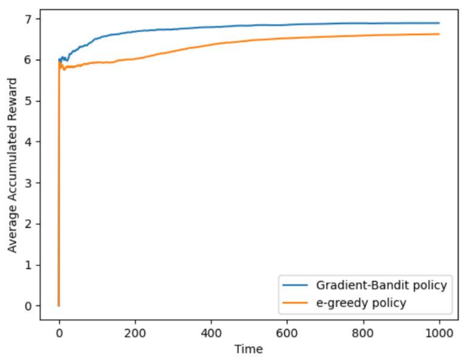


Figure 5 Average Accumulated Reward for gradient-based policy and e-greedy policy

From the plot, the gradient-based policy is better than e-greedy policy, it may result in the gradient-based policy are able to improve convergence and more effective exploration.