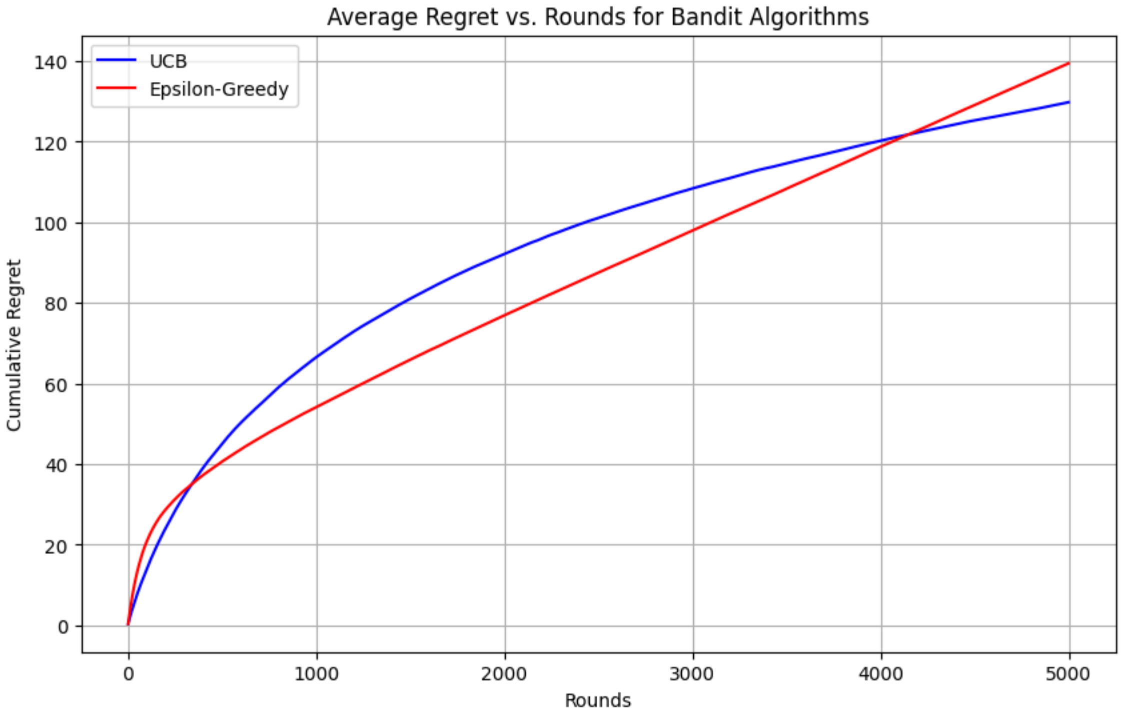
Q5

```
import numpy as np
import matplotlib.pyplot as plt


arm_probs = [0.02, 0.1, 0.25, 0.4]
num_arms = len(arm_probs)


T = 5000
num_trials = 1000

def run_ucb(num_rounds, arm_probs):
    num_arms = len(arm_probs)
    Q = np.zeros(num_arms)
    N = np.zeros(num_arms)
    regret = np.zeros(num_rounds)

    for t in range(num_rounds):
        if t < num_arms:
            arm = t
        else:
            ucb_values = Q + np.sqrt(2 * np.log(t + 1) / (N + 1e-5))
            arm = np.argmax(ucb_values)

        reward = np.random.binomial(1, arm_probs[arm])
        N[arm] += 1
        Q[arm] += (reward - Q[arm]) / N[arm]
        regret[t] = np.max(arm_probs) - arm_probs[arm]

    return np.cumsum(regret)


def run_epsilon_greedy(num_rounds, arm_probs, epsilon=0.1):
    num_arms = len(arm_probs)
    Q = np.zeros(num_arms)
    N = np.zeros(num_arms)
    regret = np.zeros(num_rounds)

    for t in range(num_rounds):
        if np.random.rand() < epsilon:
            arm = np.random.choice(num_arms)
        else:
            arm = np.argmax(Q)

        reward = np.random.binomial(1, arm_probs[arm])
        N[arm] += 1
        Q[arm] += (reward - Q[arm]) / N[arm]
        regret[t] = np.max(arm_probs) - arm_probs[arm]

    return np.cumsum(regret)


ucb_regret = np.zeros(T)
egreedy_regret = np.zeros(T)

for _ in range(num_trials):
    ucb_regret += run_ucb(T, arm_probs)
    egreedy_regret += run_epsilon_greedy(T, arm_probs)

ucb_regret /= num_trials
egreedy_regret /= num_trials


plt.figure(figsize=(10, 6))
plt.plot(ucb_regret, label='UCB', color='blue')
plt.plot(egreedy_regret, label='Epsilon-Greedy', color='red')
plt.xlabel('Rounds')
plt.ylabel('Cumulative Regret')
plt.title('Average Regret vs. Rounds for Bandit Algorithms')
plt.legend()
plt.grid()
plt.show()
```



Average Regret vs. Rounds for Bandit Algorithms
UCB
Epsilon-Greedy
Cumulative Regret
0
20
40
60
80
100
120
140
0
1000
2000
3000
4000
5000
Rounds