Exp – 2 Implementation of GREEDY AGENT

Code:

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

n_arms = 3

true_rewards = [0.2,0.5,0.75]
n_steps = 100

reward_sums = np.zeros(n_arms)
counts = np.zeros(n_arms)
estimated_rewards = np.zeros(n_arms)
reward_history = []
cumulative_reward = []

for arm in range(n_arms):
  reward = int(np.random.rand() < true_rewards[arm])
  reward_sums[arm] += reward
  counts[arm] += 1
  estimated_rewards[arm] = reward_sums[arm] / counts[arm]
  reward_history.append(reward)
  cumulative_reward.append(np.sum(reward_history))

for step in range(n_arms, n_steps):
  arm = np.argmax(estimated_rewards)
  reward = int(np.random.rand() < true_rewards[arm])

  reward_sums[arm] += reward
  counts[arm] += 1
  estimated_rewards[arm] = reward_sums[arm] / counts[arm]

  reward_history.append(reward)
  cumulative_reward.append(np.sum(reward_history))

print("\n == Final Results ==")
print("True Reward Probabilities:   ", true_rewards)
print("Estimated Reward Probabilities:",np.round( estimated_rewards,2))
print("Number of times each arm was selected:", counts)
print("Total Reward Earned:    ", int(sum(reward_history)))


plt.figure(figsize=(10,5))
plt.plot(cumulative_reward)
plt.xlabel("Steps")
plt.ylabel("Total Reward")
plt.title("Greedy Agent - Reward Over Time")
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```

Output:

```
 == Final Results ==
True Reward Probabilities:    [0.2, 0.5, 0.75]
Estimated Reward Probabilities: [0.   0.49 0.  ]
Number of times each arm was selected: [ 1. 98.  1.]
Total Reward Earned:     48
/tmp/ipython-input-5-1537682200.py:47: UserWarning: No artists with labels found to put in legend.  Note that artists whose label start with an underscore are ignored when legend() is called with no argument.
  plt.legend()
```