Algorithms Implemented

- **Monte Carlo Control (Exploring Starts initially)**
- **Monte Carlo Control (ε-greedy / Epsilon-Soft version)**
- **SARSA (On-Policy TD Learning)**
- **Q-Learning (Off-Policy TD Learning)**

Results

Algorithm	Average Return	Success Rate (%)	Final Policy (Best Actions)
MC-ES (Exploring Starts)	0.00	0%	All 0s (untrained)
Q-Learning	0.8	78-95%	[[2 2 1 0] [1 0 1 0] [2 1 1 0] [0 2 2 0]]
SARSA	0.7111	71.11-90%	[[1 2 1 0] [1 0 1 0] [2 2 1 0] [0 2 2 0]]
MC-ES (Epsilon-Soft)	0.78	78-90%	

: Different runs and varying gamma (discount factor) values led to different outcomes here are some:

```
MC-ES -> Avg Success Rate: 0.00% ± 0.00%
MC-ES Policy (Last Run)
[[0 0 0 0]
 [0 0 0 0]
 [0 0 0 0]
 [0 0 0 0]]

Q-Learning -> Avg Success Rate: 100.00% ± 0.00%
Q-Learning Policy (Last Run)
[[2 2 1 0]
 [3 0 1 0]
 [2 2 1 0]
 [0 2 2 0]]

SARSA -> Avg Success Rate: 100.00% ± 0.00%
SARSA Policy (Last Run)
[[1 2 1 0]
 [1 0 1 0]
 [2 2 1 0]
 [0 2 2 0]]
```

```
(base) tazmeen@afroz:~$ python3 rl2.py

--- Run 1/1 ---

Training with Q-Learning...
Q-Learning Episode 0/10000
Q-Learning Episode 1000/10000
Q-Learning Episode 2000/10000
Q-Learning Episode 3000/10000
Q-Learning Episode 4000/10000
Q-Learning Episode 5000/10000
Q-Learning Episode 6000/10000
Q-Learning Episode 7000/10000
Q-Learning Episode 8000/10000
Q-Learning Episode 9000/10000
Q-Learning Average Return: 0.7111
Success rate: 71.11%

Training with SARSA...
SARSA Episode 0/10000
SARSA Episode 1000/10000
SARSA Episode 2000/10000
SARSA Episode 3000/10000
SARSA Episode 4000/10000
SARSA Episode 5000/10000
SARSA Episode 6000/10000
SARSA Episode 7000/10000
SARSA Episode 8000/10000
SARSA Episode 9000/10000
SARSA Average Return: 0.7111
Success rate: 71.11%

Training with Monte Carlo ES...
MC-ES Episode 0/10000
MC-ES Episode 1000/10000
MC-ES Episode 2000/10000
MC-ES Episode 3000/10000
MC-ES Episode 4000/10000
MC-ES Episode 5000/10000
MC-ES Episode 6000/10000
MC-ES Episode 7000/10000
MC-ES Episode 8000/10000
MC-ES Episode 9000/10000
MC-ES Final Success Rate: 98.50%

MC-ES -> Avg Success Rate: 100.00% ± 0.00%
MC-ES Policy (Last Run)
[[2 2 1 0]
 [1 0 1 0]
 [2 1 1 0]
 [0 2 2 0]]
```

```
SARSA Episode 8000/10000
SARSA Episode 9000/10000
SARSA Average Return: 0.7111
Success rate: 71.11%

Training with Monte Carlo ES...
MC-ES Episode 0/10000
MC-ES Episode 1000/10000
MC-ES Episode 2000/10000
MC-ES Episode 3000/10000
MC-ES Episode 4000/10000
MC-ES Episode 5000/10000
MC-ES Episode 6000/10000
MC-ES Episode 7000/10000
MC-ES Episode 8000/10000
MC-ES Episode 9000/10000
MC-ES Final Success Rate: 98.50%

MC-ES -> Avg Success Rate: 100.00% ± 0.00%
MC-ES Policy (Last Run)
[[2 2 1 0]
 [1 0 1 0]
 [2 1 1 0]
 [0 2 2 0]]

Q-Learning -> Avg Success Rate: 100.00% ± 0.00%
Q-Learning Policy (Last Run)
[[2 2 1 0]
 [1 0 1 0]
 [2 2 1 0]
 [0 2 2 0]]

SARSA -> Avg Success Rate: 100.00% ± 0.00%
SARSA Policy (Last Run)
[[2 2 1 0]
 [1 0 1 0]
 [2 2 1 0]
 [0 2 2 0]]

Creating animation for MC-ES...
Success rate over 10 episodes: 100.0%
Animation saved to MC-ES_animation.gif

Creating animation for Q-Learning...
Success rate over 10 episodes: 100.0%
Animation saved to Q-Learning_animation.gif

Creating animation for SARSA...
Success rate over 10 episodes: 100.0%
Animation saved to SARSA_animation.gif
```

```
===== RESULTS SUMMARY =====
Number of episodes: 10000
Environment: FrozenLake-v1 (is_slippery=False)

Monte Carlo ES:
  Average Return: 0.0000
  Success Rate: 0.00%
  Total Time: 89.82 seconds

Monte Carlo ε-Soft:
  Average Return: 0.3529
  Success Rate: 35.29%
  Total Time: 58.36 seconds

Q-Learning:
  Average Return: 0.6989
  Success Rate: 69.89%
  Total Time: 5.28 seconds

SARSA:
  Average Return: 0.0000
  Success Rate: 0.00%
  Total Time: 15.45 seconds

(base) tazmeen@afroz:~$
```

```
===== RESULTS SUMMARY =====
Number of episodes: 10000
Environment: FrozenLake-v1 (is_slippery=False)

Monte Carlo ES:
  Average Return: 0.0000
  Success Rate: 0.00%
  Total Time: 74.12 seconds

Monte Carlo ε-Soft:
  Average Return: 0.4298
  Success Rate: 42.98%
  Total Time: 44.52 seconds

Q-Learning:
  Average Return: 0.0962
  Success Rate: 9.62%
  Total Time: 12.34 seconds

SARSA:
  Average Return: 0.0000
  Success Rate: 0.00%
  Total Time: 16.68 seconds

(base) tazmeen@afroz:~$
```

```
===== RESULTS SUMMARY =====
Number of episodes: 10000
Environment: FrozenLake-v1 (is_slippery=False)

Monte Carlo ES:
  Average Return: 0.0000
  Success Rate: 0.00%
  Total Time: 60.24 seconds

Monte Carlo ε-Soft:
  Average Return: 0.7829
  Success Rate: 78.29%
  Total Time: 29.53 seconds

Q-Learning:
  Average Return: 0.0000
  Success Rate: 0.00%
  Total Time: 11.55 seconds

SARSA:
  Average Return: 0.7587
  Success Rate: 75.87%
  Total Time: 2.15 seconds

(base) tazmeen@afroz:~$ gedit test8.py
(base) tazmeen@afroz:~$ gedit test9.py
```

(The original script did not include animation code. Animation was generated separately using code provided by GPT)

```
# Function to create and save animation
def create_animation(frames, filename="agent_animation.gif", fps=4):
        fig, ax = plt.subplots(figsize=(5, 5))
        plt.tight_layout()
        plt.axis('off')

        # Create animation
        patch = ax.imshow(frames[0])

        def animate(i):
        patch.set_array(frames[i])
        return [patch]

        anim = animation.FuncAnimation(
        fig, animate, frames=len(frames), interval=1000//fps)

        # Save animation
        anim.save(filename, writer=PillowWriter(fps=fps))
        plt.close()

        print(f"Animation saved to {filename}")
        return filename
```