Mehmet Ali Pelit
21113622

1 SARSA (State-Action-Reward-State-Action)

1.1 Mathematical Definition

SARSA is an on-policy reinforcement learning algorithm defined by the following update rule:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left(r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\right)$$

1.2 Code Design

- Initialize the Q-values table (Q) for all state-action pairs.
- For each episode:
  - Choose an action a_t using an epsilon-greedy policy.
  - Observe reward r_t and next state s_{t+1}.
  - Choose next action a_{t+1}.
  - Update Q-values using the SARSA update rule.

1.3 Implementation Issues

- The choice of epsilon for the epsilon-greedy policy can greatly affect learning.
- SARSA converges under certain conditions; careful tuning of parameters is necessary for stability.

2 Q-Learning

2.1 Mathematical Definition

Q-Learning is an off-policy reinforcement learning algorithm defined by:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left(r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right)$$

2.2 Code Design

- Similar initialization of the Q-values table.
- For each episode:
  - Find an action a_t which makes Q-value max
  - Observe reward r_t and next state s_{t+1}.
  - Update Q-values using the Q-Learning update rule.

2.3 Implementation Issues

- **Off-Policy Learning:** Care must be taken when the policy used to generate behavior differs from the policy being evaluated.
- Requires sufficient exploration of the state space to ensure convergence to the optimal policy.

Mehmet Ali Pelit
21113622

3 Expected SARSA

3.1 Mathematical Definition

Expected SARSA modifies the SARSA update rule by using the expected value of the next state's Q-values:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left(r_t + \gamma \sum_a \pi(a|s_{t+1}) Q(s_{t+1}, a) - Q(s_t, a_t) \right)$$

where $\pi(a|s)$ is the policy used to select actions.

3.2 Code Design

- Similar initialization and episode loop as SARSA.
- Calculate the expected Q-value based on the policy when updating Q-values.

3.3 Implementation Issues

- The policy needs to be represented and used to compute expected values accurately.
- The expected value calculation can be computationally intensive, especially in large action spaces.
- Balancing exploration and exploitation remains essential, as in SARSA.

4 Double Q-Learning

4.1 Mathematical Definition

Double Q-Learning aims to reduce the overestimation bias by maintaining two separate Q-value functions:

$$Q_1(s_t, a_t) \leftarrow Q_1(s_t, a_t) + \alpha \left(r_t + \gamma Q_2(s_{t+1}, \text{argmax}_a Q_1(s_{t+1}, a)) - Q_1(s_t, a_t) \right)$$

$$Q_2(s_t, a_t) \leftarrow Q_2(s_t, a_t) + \alpha \left(r_t + \gamma Q_1(s_{t+1}, \text{argmax}_a Q_2(s_{t+1}, a)) - Q_2(s_t, a_t) \right)$$

where Q_2 is the second Q-value function.

4.2 Code Design

- Initialize two Q-value tables, Q_1 and Q_2.
- For each episode:
  - Randomly pick a Q-table, Q_1 or Q_2
  - Choose action a_t which make the relevant Q-table value maximum for that state.
  - Observe reward r_t and next state s_{t+1}.
  - Update one of the Q-value tables using the update rule and alter nate which table to update.

4.3 Implementation Issues

- Ensuring that both Q-value tables are updated appropriately can complicate implementation.
- Maintaining two Q-tables increases memory usage.
- Must ensure that both Q-tables are adequately explored to achieve convergence.

Mehmet Ali Pelit
21113622

Results Analysis

4.4 Task 1

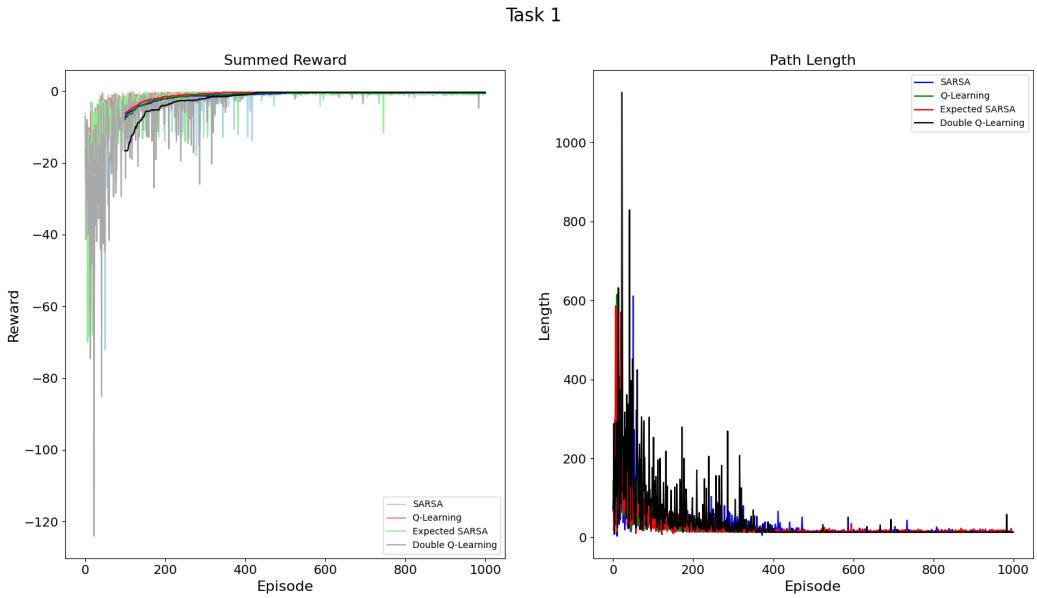

Figure 1: Task 1 Experiment Plots

Text Output:

```
All experiments complete
Experiment Setup:
 - episodes:1000
 - sim_speed:0.01

[SARSA on Task 1] : SARSA : max-rew=-0.300 med-100=-0.500 var-100=0.047 max-episode-len=611.0
[Q-Learning on Task 1] : Q-Learning : max-rew=-0.300 med-100=-0.300 var-100=0.003 max-episode-len=615.0
[Expected SARSA on Task 1] : Expected SARSA : max-rew=-0.300 med-100=-0.500 var-100=0.065 max-episode-len=586.0
[Double Q-Learning on Task 1] : Double Q-Learning : max-rew=-0.300 med-100=-0.300 var-100=0.203 max-episode-len=1127.0
```

Mehmet Ali Pelit
21113622

4.5 Task 2

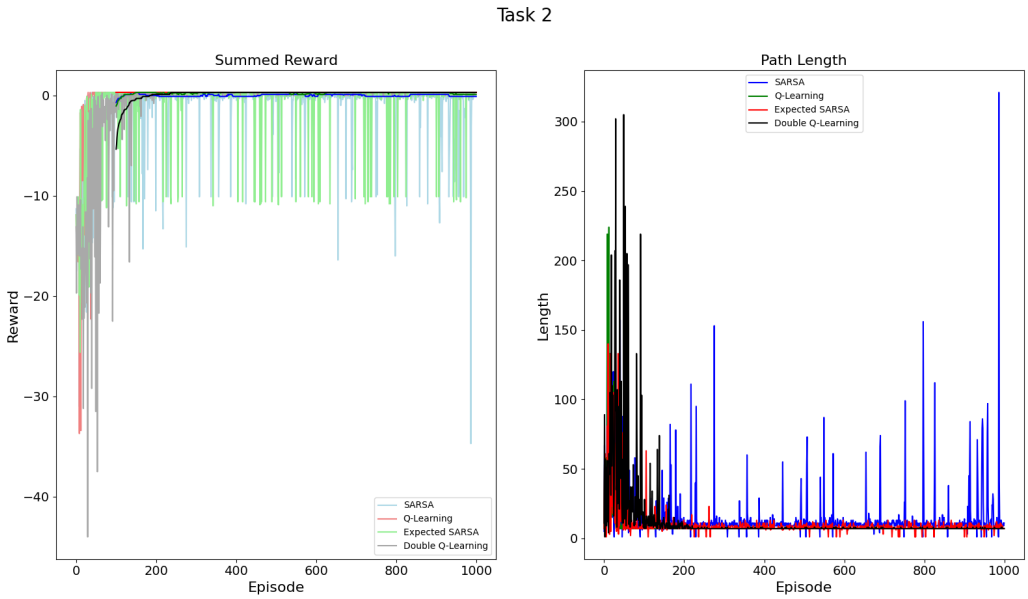

Figure 2: Task 2 Experiment Plots

Text Output:

```
All experiments complete
Experiment Setup:
 - episodes:1000
 - sim_speed:0.01

[SARSA on Task 2] : SARSA : max-rew=0.300 med-100=-0.100 var-100=21.184 max-episode-len=321.0
[Q-Learning on Task 2] : Q-Learning : max-rew=0.300 med-100=0.300 var-100=0.003 max-episode-len=224.0
[Expected SARSA on Task 2] : Expected SARSA : max-rew=0.300 med-100=0.100 var-100=7.153 max-episode-len=187.0
[Double Q-Learning on Task 2] : Double Q-Learning : max-rew=0.300 med-100=0.300 var-100=0.004 max-episode-len=305.0
```

Mehmet Ali Pelit
21113622

4.6 Task 3

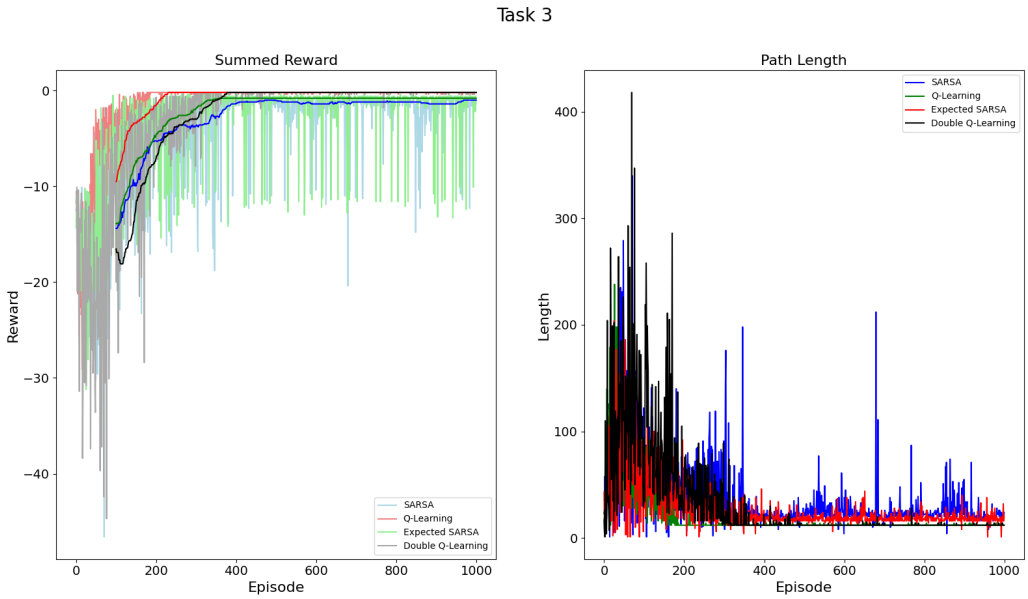

Figure 3: Task 3 Experiment Plots

Text Output:

```
All experiments complete
Experiment Setup:
 - episodes:1000
 - sim_speed:0.01

[SARSA on Task 3] : SARSA : max-rew=-0.400 med-100=-1.000 var-100=4.741 max-episode-len=340.0
[Q-Learning on Task 3] : Q-Learning : max-rew=-0.200 med-100=-0.200 var-100=0.003 max-episode-len=238.0
[Expected SARSA on Task 3] : Expected SARSA : max-rew=-0.200 med-100=-0.800 var-100=9.760 max-episode-len=217.0
[Double Q-Learning on Task 3] : Double Q-Learning : max-rew=-0.200 med-100=-0.200 var-100=0.004 max-episo
```

5 Conclusion

Since it is almost midnight, I couldn't find enough time to make a conclusion and deduction from the experimental results. I will send the updated version again tomorrow. (Hopefully, it will be accepted! Happy Thanksgiving!)