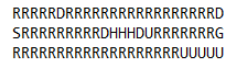
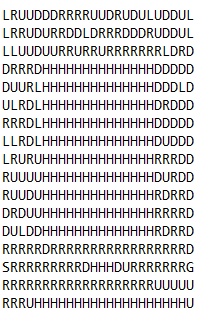
**Result**

Based on the output of three algorithm, I found out that the Q-learning always can find the optimal policy towards the goal. In this grid world, it always goes to right, when reaching the edge of the hole, it goes around the hole and directly goes toward the goal. Like this: 

For other parts, there may not be a path to the goal, and may occur a loop path, just stuck there.



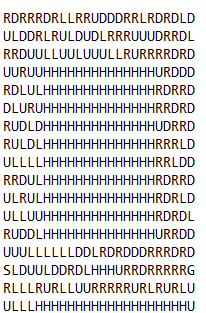
In the other parts of the path, there still can be loop path, may stuck there. I ran a lot of times for the Q-learning algorithm, almost every time it can get optimal path for the part around start point to the goal point like the first picture shows.

Besides, Q-learning runs in a second for every episodes reduction methods.

However for the SARSA algorithm, it is not guaranteed to find the optimal policy, it tends to avoid the holes, and go around the holes rather than going right toward the goal like the policy comes from Q-Learning algorithm.

It may stuck somewhere sometimes, stuck in a loop path forever until the “3|S|” limited condition break the loop for that episode.

Following graph is one sample run of the SARSA algorithm. There is not a optimal policy in this run. And it takes longer to run the algorithm than Q Learning. As we can see like this:, it is a loop path.

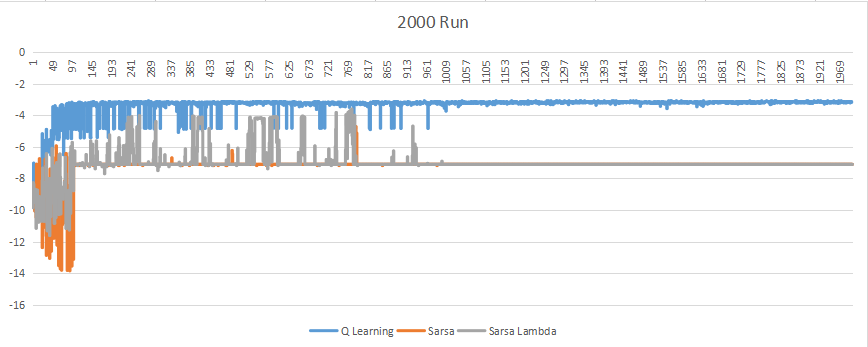


I ran a lot of times for SARSA algorithm, it is really hard to find a path to the goal.

For the SARSA lambda, the policy it produces are similar to the one of SARSA. It is really hard to find the path to the goal. I ran several times and I didn’t see the path. It may stuck somewhere in a loop path like what happened in SARSA algorithm. It also takes longer to run than Q Learning.

Analysis

The following diagrams shows the reward gained for 2000 episodes runs. I used log(reward ) to scale the rewards.

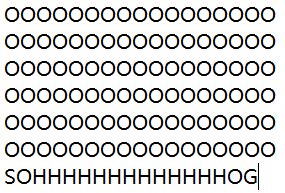
In the beginning stage, because both three algorithm use almost greedy method to take an action. Therefore, when the epsilon is not small enough in the beginning, it is easily to get an random action for the agent. The rewards gained are severely changeable. Additionally, for the reward gained by SARSA and SARSA lambda are lower than Q learning in this stage.

After certain episodes which makes the action not that randomly (when epsilon less than 0.1, which means 0.9 chance to find greedy action and 0.1 chance still find random action. In 2000 runs, the episodes is 100). The rewards tends to be stable, for the Q learning, the value changes around -24，both the SARSA and SARSA lambda changes around -1224. The reward still is changeable but slightly. Because in this domain, there are icy surface, the agent may slip to the hole even though it already find the best path. So the rewards gained in this stage is still changeable. .

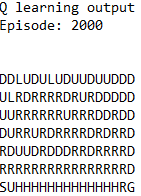
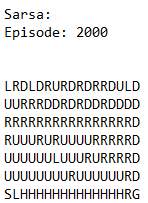
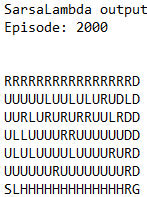
The Q learning algorithm is Off-Policy Methods which update the values we learn based on the best actions only. It pays no attention to the actual policy being followed. For every next state, it takes the action which has maximum Q value. It ignores rewards and outcomes that come from any of the possible bad actions we take when exploring, which means the hole in this grid world. So it always can find the optimal policy for the problem, ignoring the effect of holes besides the path.

For SARSA and SARSA lambda algorithm, they are both on-policy updating method, we are not basing our value calculation on the best possible policy. Instead, we are basing it on our learning policy, which means the values that we base our updates and choices on will combine the values that we get from Almost-Greedy Policies : 1. greedy action selection for exploitation 2. random actions, consequently, it won’t always take the best action even though we already know which one is the best, we still may take random actions sometimes. They learn the safe path, it actually receives a higher average reward per trial than Q-Learning even though it does not walk the optimal path. So, we sometimes acted in a non-optimal way, and may even can’t find the policy for the problem.

I also did some experiment, I changed the problem domain like this:

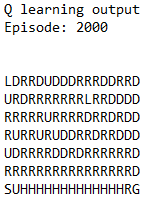
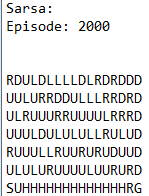
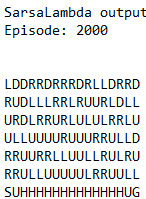


When I Set ALPHA to 0.5，the result is like following:

The Q learning always find the optimal path for the problem, SARSA find the safe path away from the hole, but the SARSA Lambda find the safest path, which along the upper bound path to the goal.

When I set ALPHA to 0.9，the result is like following:

Both SARSA and SARSA Lambda goes toward the most upper bound, which is the safest path. But there can be a loop path.

I tried some other ALPHA value as well, and I found out if the ALPHA is smaller, the chance to find a path is higher. And the path SARSA finds is closer to the upper bound.

Based on these, I think the value of ALPHA and problem domain have effects on whether the algorithm can find the optimal path. If problem domain is big, it is hard to find a path, and takes much longer time to run the algorithm.