**Machine Learning**

COMP5611M

**Coursework2: Reinforcement Learning**

Daolin Sheng ml192ds

Experiments

1. All of experiment under the same parameters except for the epsilon greedy(wpsoffice-greedy):

(1). learning\_rate = 0.01

(2). gamma = 0.99

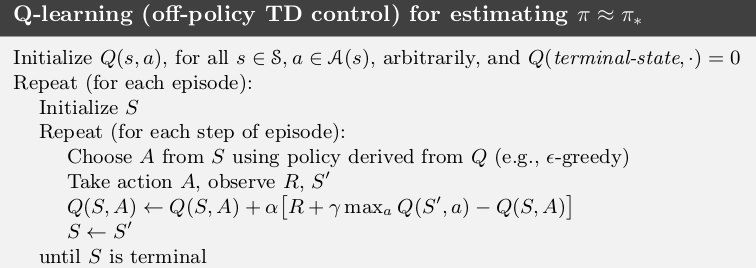
(3). maxStepsPerEpisode = 2500 (max number of steps possible in a single episode)

(4). nbOfTrainingEpisodes = 1100

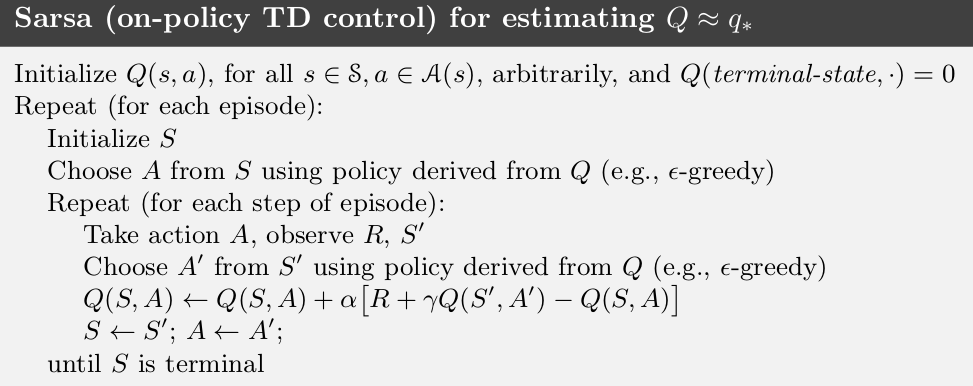
2. The Q-learning and SARSA algorithm use the same policy: wpsoffice-greedy.

3. In all experiments, I will use the function ‘updateEpsilon(self, episode\_counter)’ to upadte the epsilon value from the lager to lower(from explorartion to exploitation).

4. The Q-learing algorithm:

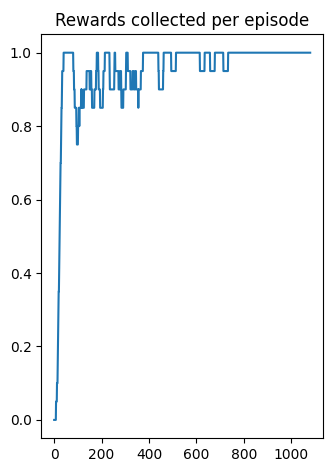
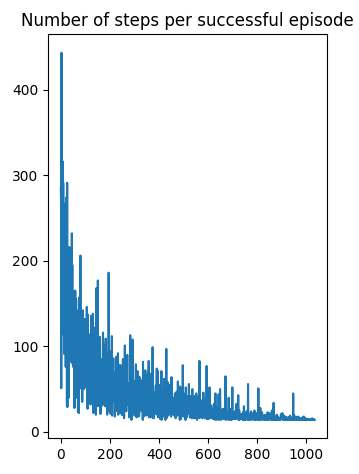


5. The Sarsa algorithm:

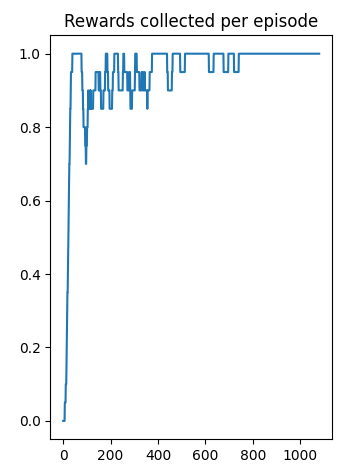
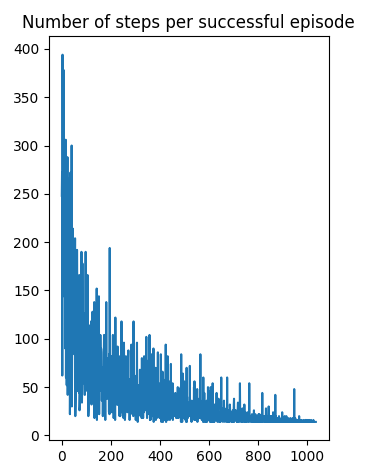


\* *References : the two Images comes from the https://www.google.com.*

**1, Q-learning: wpsoffice = 0.8:**

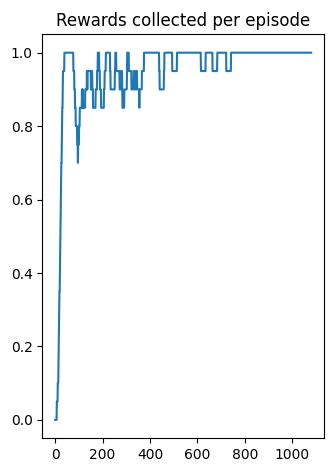
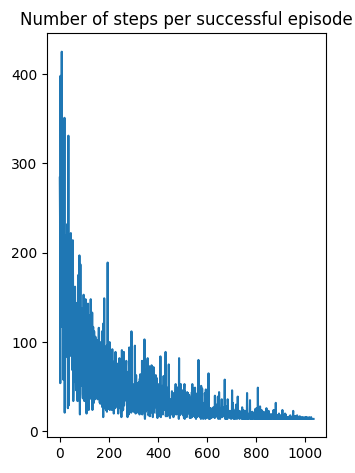


**2, SARSA: wpsoffice = 0.8:**

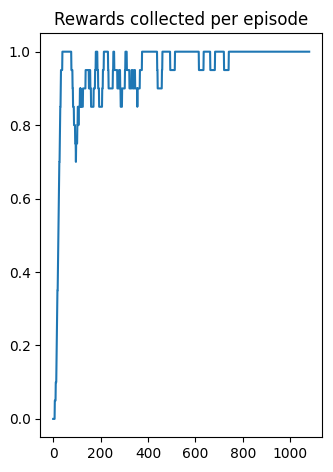
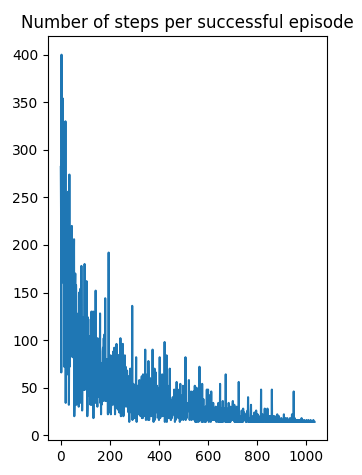


At the same epsilon value 0.8, Q-learning is an off-policy TD control policy , and SARSA is an on-policy TD control method. By comparing plots of these two algorithms, we can the truth that the average of the number of steps per successful episode of Q-learning is less than that of SARSA due to Q-learing always attempts to follow the optiomal path which is the shorted one. By way of contrast, the sarsa algorithm will converge to a much safer route that keeps it well away from the cliffff, even though it takes longer.

**3, Q-learning: wpsoffice = 0.1:**



**4, SARSA: wpsoffice = 0.1:**



At the same epsilon value 0.1 (lower than 0.8), which means Q-learning and Sarsa use the higher exploitation in this experiment. So, the avarege number of the number of steps per successful episode of these algorithms are less than previous two experiments in epsilon 0.8.