COMP3702 Artificial Intelligence (Semester 2, 2022)
Assignment 3: HEXBOT Reinforcement Learning

Name: Zhiyu Chen

Student ID: 45714568

Student email: s4571456@student.uq.edu.au

Note: Please edit the name, student ID number and student email to reflect your identity and **do not modify the design or the layout in the assignment template**, including changing the paging.

Question 1 (Complete your full answer to Question 1 on the remainder of page 1)

Q-learning is closely related to the Value Iteration algorithm for Markov decision processes.

a) (5 marks) Describe two key similarities between Q-learning and Value Iteration.

b) (5 marks) Give one key difference between Q-learning and Value Iteration.

a)

1. Both of them are based on evaluating q values for each state in the environment.

2. Both of them update q values asynchronously.

b) There is no transition function or transition probability in Q-learning, so it uses the experience. But Value Iteration has transition function and probability, so it evaluates the exact known values. In other words, Q learning is model-free while Value Iteration is based on model.

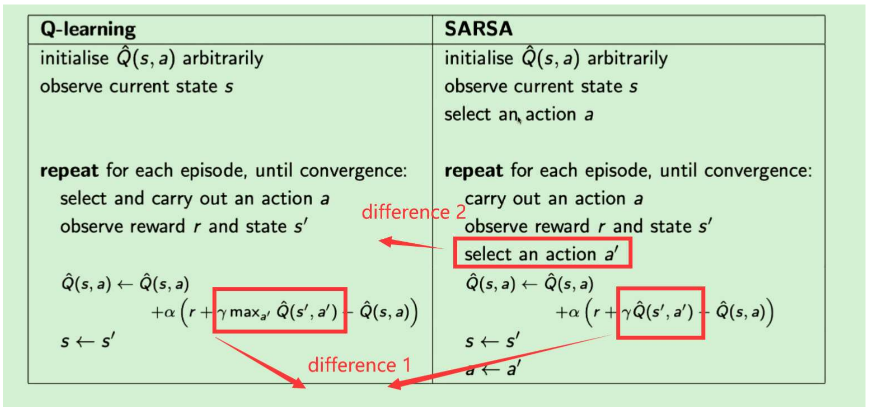
Question 2 (Complete your full answer to Question 2 on page 2)

a) (5 marks) Explain the difference between off-policy and on-policy reinforcement learning algorithms.

Explain where this difference is represented in the Q-learning and SARSA algorithms.

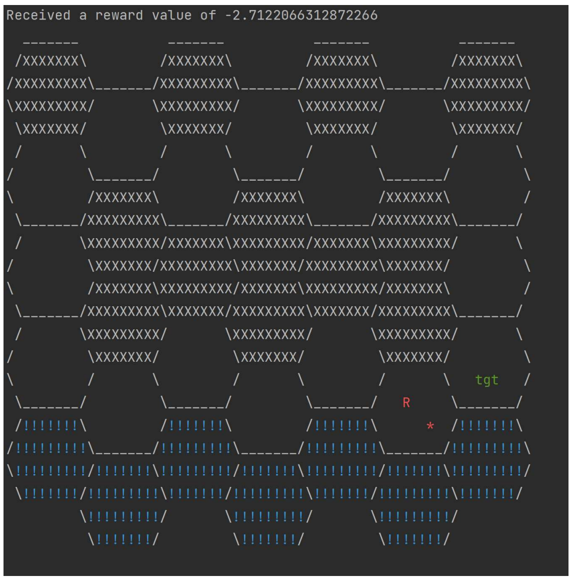
Q-learning (off policy) always ideally chooses the maximum q value for next state among all possible q values in that state, but the next action taken might not be the one with corresponding maximum q value. SARSA (on policy) chooses the real action to be take for the next state and evaluate q values based on the real action.

In my code algorithms, Q learning compares the q value of four actions in next state and choose the one with highest q value for that state. After that, q learning update q(s,a) using that maximum q value although that action might not be taken in next state. SARSA algorithm decides which action to take for next state before updating q(s,a). If agent decide to explore rather than exploit, it would use the q value of exploration action to update q(s,a).

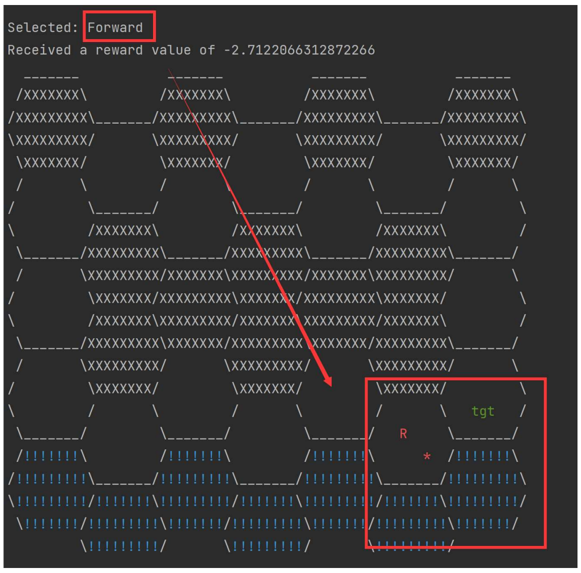


Q-learning	**SARSA**
initialise $\hat{Q}(s,a)$ arbitrarily	initialise $\hat{Q}(s,a)$ arbitrarily
observe current state s	observe current state s
	select an action a
repeat for each episode, until convergence:	**repeat** for each episode, until convergence:
select and carry out an action a	carry out an action a
observe reward r and state s'	observe reward r and state s'
	$\boxed{\text{select an action } a'}$
$\hat{Q}(s,a) \leftarrow \hat{Q}(s,a)$	$\hat{Q}(s,a) \leftarrow \hat{Q}(s,a)$
$+\alpha\left(r + \boxed{\gamma \max_{a'} \hat{Q}(s',a')} - \hat{Q}(s,a)\right)$	$+\alpha\left(r + \boxed{\gamma \hat{Q}(s',a')} - \hat{Q}(s,a)\right)$
$s \leftarrow s'$	$s \leftarrow s'$
	$a \leftarrow a'$

b) (5 marks) How does the difference between off-policy and on-policy algorithms affect the way in which Q-learning and SARSA solve test cases ex3.txt and ex4.txt? If you were unable to solve these test cases, it is sufficient to answer with reference to what you think would happen, based on the set up described in the test case files.



```
Received a reward value of -2.7122066312872266

  -------           -------           -------           -------
 /XXXXXXX\         /XXXXXXX\         /XXXXXXX\         /XXXXXXX\
/XXXXXXXXX\_______/XXXXXXXXX\_______/XXXXXXXXX\_______/XXXXXXXXX\
\XXXXXXXXX/       \XXXXXXXXX/       \XXXXXXXXX/       \XXXXXXXXX/
 \XXXXXXX/         \XXXXXXX/         \XXXXXXX/         \XXXXXXX/
 /       \         /       \         /       \         /       \
/         \_______/         \_______/         \_______/         \
\         /XXXXXXX\         /XXXXXXX\         /XXXXXXX\         /
 \_______/XXXXXXXXX\_______/XXXXXXXXX\_______/XXXXXXXXX\_______/
 /       \XXXXXXXXX/XXXXXXX\XXXXXXXXX/XXXXXXX\XXXXXXXXX/       \
/         \XXXXXXX/XXXXXXXXX\XXXXXXX/XXXXXXXXX\XXXXXXX/         \
\         /XXXXXXX\XXXXXXXXX/XXXXXXX\XXXXXXXXX/XXXXXXX\         /
 \_______/XXXXXXXXX\XXXXXXX/XXXXXXXXX\XXXXXXX/XXXXXXXXX\_______/
 /       \XXXXXXXXX/       \XXXXXXXXX/       \XXXXXXXXX/       \
/         \XXXXXXX/         \XXXXXXX/         \XXXXXXX/         \
\         /       \         /       \         /       \   tgt   /
 \_______/         \_______/         \_______/   R     \_______/
 /!!!!!!!\         /!!!!!!!\         /!!!!!!!\      *  /!!!!!!!\
/!!!!!!!!!\_______/!!!!!!!!!\_______/!!!!!!!!!\_______/!!!!!!!!!\
\!!!!!!!!!/!!!!!!!\!!!!!!!!!/!!!!!!!\!!!!!!!!!/!!!!!!!\!!!!!!!!!/
 \!!!!!!!/!!!!!!!!!\!!!!!!!/!!!!!!!!!\!!!!!!!/!!!!!!!!!\!!!!!!!/
         \!!!!!!!!!/       \!!!!!!!!!/       \!!!!!!!!!/
          \!!!!!!!/         \!!!!!!!/         \!!!!!!!/
```



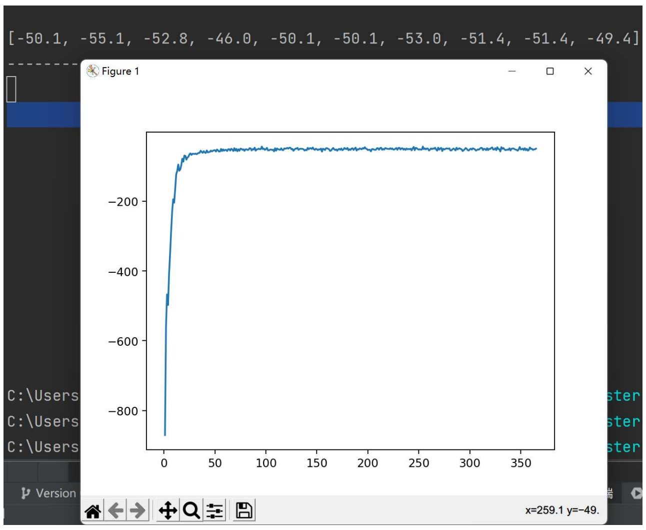
```
Selected: Forward
Received a reward value of -2.7122066312872266

  -------           -------           -------           -------
 /XXXXXXX\         /XXXXXXX\         /XXXXXXX\         /XXXXXXX\
/XXXXXXXXX\_______/XXXXXXXXX\_______/XXXXXXXXX\_______/XXXXXXXXX\
\XXXXXXXXX/       \XXXXXXXXX/       \XXXXXXXXX/       \XXXXXXXXX/
 \XXXXXXX/         \XXXXXXX/         \XXXXXXX/         \XXXXXXX/
 /       \         /       \         /       \         /       \
/         \_______/         \_______/         \_______/         \
\         /XXXXXXX\         /XXXXXXX\         /XXXXXXX\         /
 \_______/XXXXXXXXX\_______/XXXXXXXXX\_______/XXXXXXXXX\_______/
 /       \XXXXXXXXX/XXXXXXX\XXXXXXXXX/XXXXXXX\XXXXXXXXX/       \
/         \XXXXXXX/XXXXXXXXX\XXXXXXX/XXXXXXXXX\XXXXXXX/         \
\         /XXXXXXX\XXXXXXXXX/XXXXXXX\XXXXXXXXX/XXXXXXX\         /
 \_______/XXXXXXXXX\XXXXXXX/XXXXXXXXX\XXXXXXX/XXXXXXXXX\_______/
 /       \XXXXXXXXX/       \XXXXXXXXX/       \XXXXXXXXX/       \
/         \XXXXXXX/         \XXXXXXX/         \XXXXXXX/         \
\         /       \         /       \         /       \   tgt   /
 \_______/         \_______/         \_______/   R     \_______/
 /!!!!!!!\         /!!!!!!!\         /!!!!!!!\      *  /!!!!!!!\
/!!!!!!!!!\_______/!!!!!!!!!\_______/!!!!!!!!!\_______/!!!!!!!!!\
\!!!!!!!!!/!!!!!!!\!!!!!!!!!/!!!!!!!\!!!!!!!!!/!!!!!!!\!!!!!!!!!/
 \!!!!!!!/!!!!!!!!!\!!!!!!!/!!!!!!!!!\!!!!!!!/!!!!!!!!!\!!!!!!!/
         \!!!!!!!!!/       \!!!!!!!!!/       \!!!!!!!!!/
```

The left picture is the path chosen by Q-learning, the agent just directly heads to the targets in the bottom path, and it collides with the hazard many times. However, SARSA algorithm teach the agent to take similar actions in bottom path like that taught in Q-learning, but more unreasonable actions were taken by agents like spinning around during the process or suddenly go back in the process. This is different from my expect. In my opinion, agents should take the upper path which would help avoid higher penalty. I think this is because the penalty for colliding into hazard is too low shown in the right picture in visualized record so that agent would ignore the risk.

Question 3 (Complete your full answer to Question 3 on page 3)

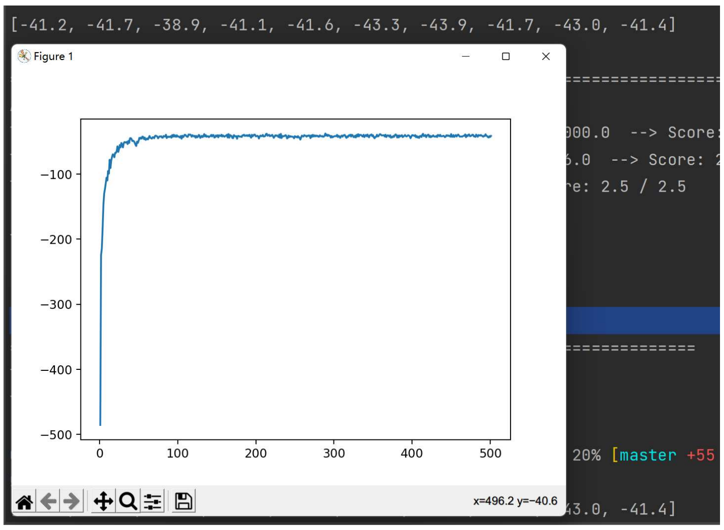
Note: my submitted code sets the agent to a random position after it reach the target. In order to get a complete reward for question3, 4, I changed my code to set the agent to initial position after it reach the target. So, the code might be a little different, but the logic is the same. And plot is generated using matplotlib which is not in the submitted version in gradescope.

a) The list with 10 numbers is the average reward for last 500 episode divided into 10 group with 50 episode each.



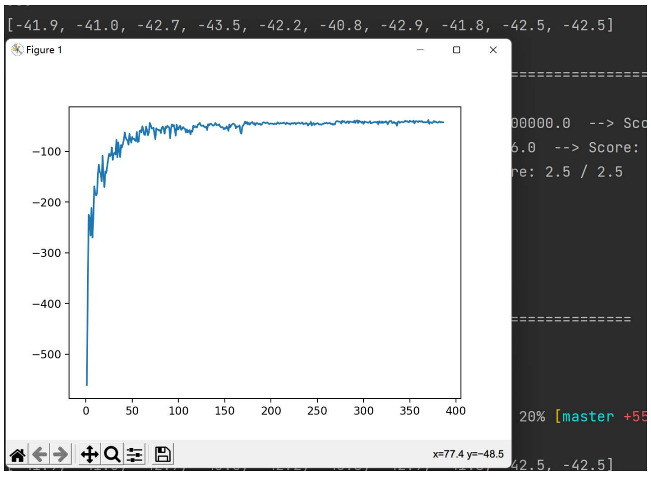
```
[-50.1, -55.1, -52.8, -46.0, -50.1, -50.1, -53.0, -51.4, -51.4, -49.4]
C:\Users
C:\Users
C:\Users
```

Figure 1 learning rate = 0.01



```
[-41.2, -41.7, -38.9, -41.1, -41.6, -43.3, -43.9, -41.7, -43.0, -41.4]
000.0  --> Score:
6.0  --> Score: 2
re: 2.5 / 2.5
20% [master +55
43.0, -41.4]
```

Figure 2 learning rate = 0.1



```
[-41.9, -41.0, -42.7, -43.5, -42.2, -40.8, -42.9, -41.8, -42.5, -42.5]
00000.0  --> Sco
6.0  --> Score:
re: 2.5 / 2.5
20% [master +55
42.5, -42.5]
```

Figure 3 learning rate = 0.8

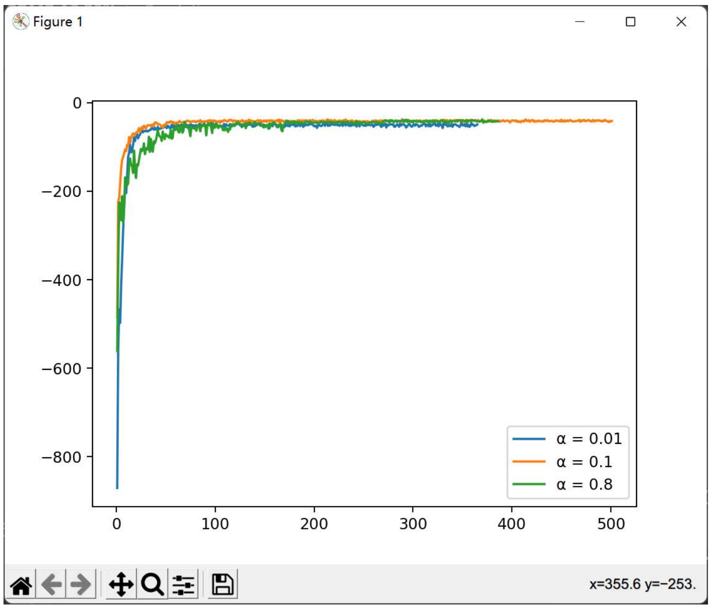


Figure4 result in one plot

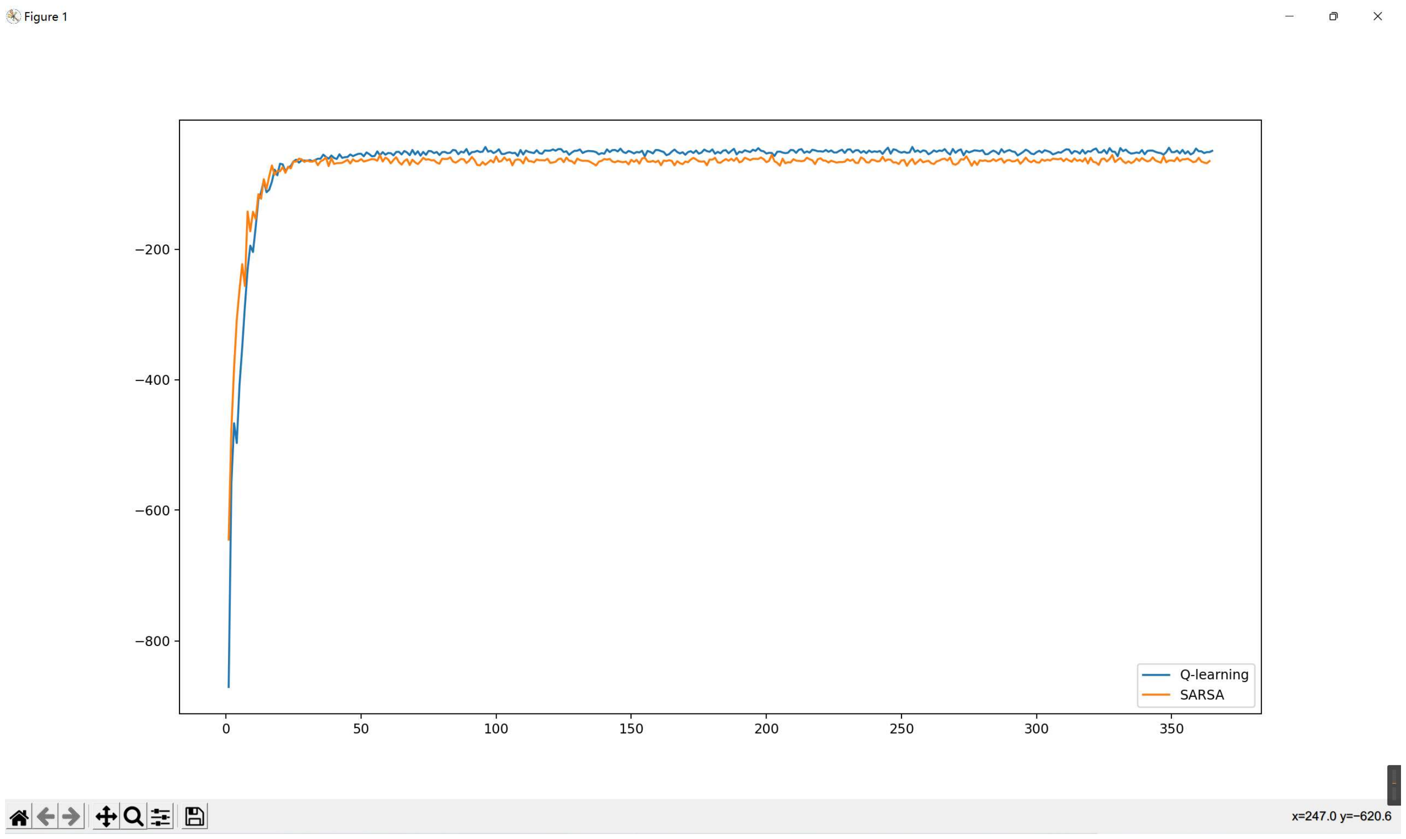
b)
$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right)$$

The learning rate affects the weight of the difference between learning outcome and the past experience. If learning rate is very high, the q_value would reach a close range of final value faster but keeps changing in a range near the converge value because the difference is overweighed for each sample. As can be seen from figure 4 the line with α = 0.8 is the most unstable one.

If the learning rate is very low, it would take more time to reach the close range of final value, but the values would be more stable and closer to the converge value as former samples would have more weight. As can be seen in figure 4, the line with α = 0.1 gets faster to converged value than the line with α = 0.01

Question 4 (Complete your full answer to Question 4 on page 4)

a)



b)

Both two algorithms get stabilised around 25*50 = 750 episodes, SARSA algorithm gets higher rewards before getting stabilised, which means agent makes fewer mistakes like colliding with hazard during the exploration.

Q-learning takes greedy strategy to select actions, so it gets closer to the converge faster.