CS 533

Assignment 3

Distributed Temporal Difference Learning

Name:

Jui-Hung Lu (933-293-709)

Chi Wen (933-276-677)

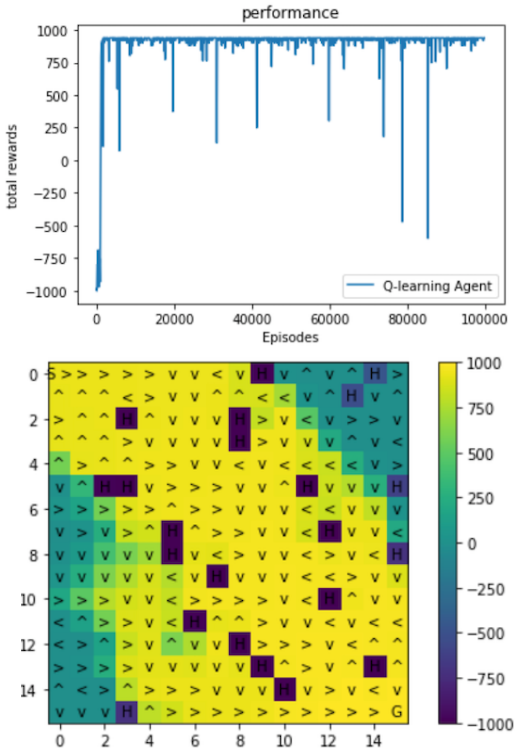
1. Provide the learning curves for the above experiments. Clearly label the curves by the parameters used.

Map_16x16 (learning episodes = 100,000, test_interval = 100)

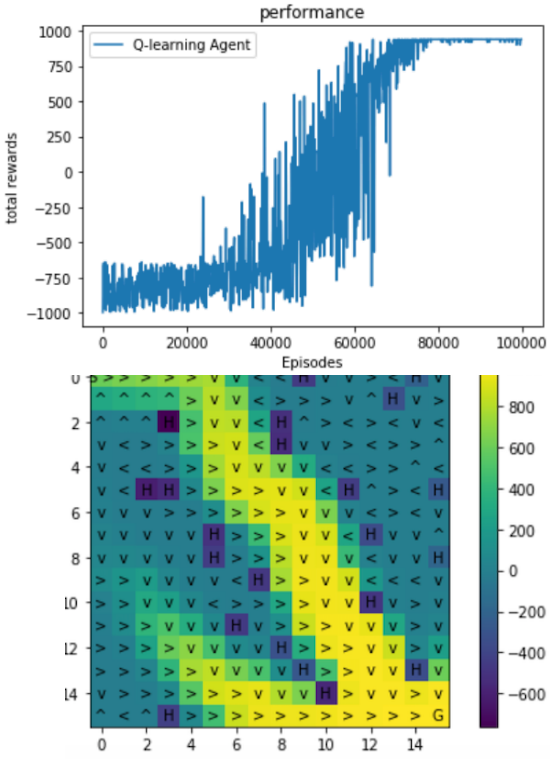
- **Q-learning agent**

(epsilon = 0.3)

learning rate = 0.1, time = 98.7092294

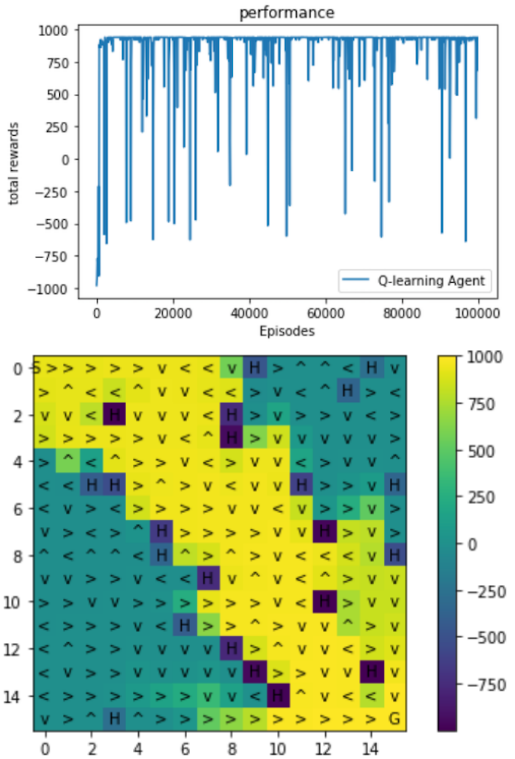


learning rate = 0.001, time = 461.642726182

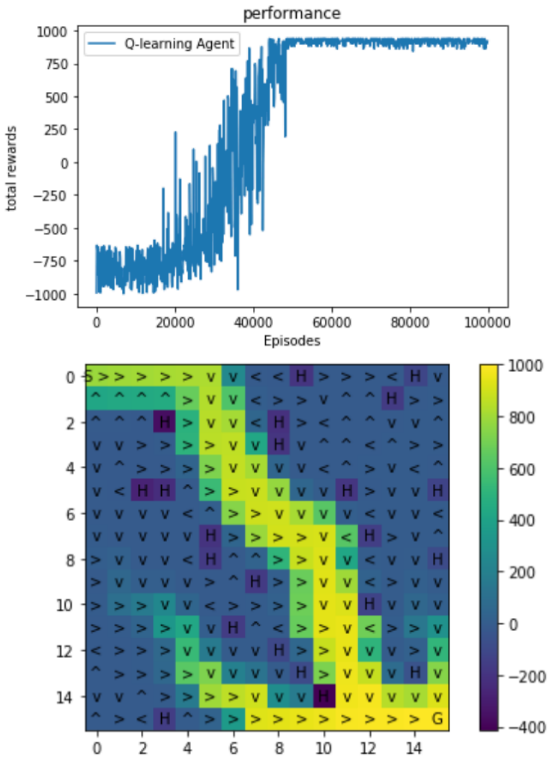


(epsilon = 0.05)

learning rate = 0.1, time = 108.884003639



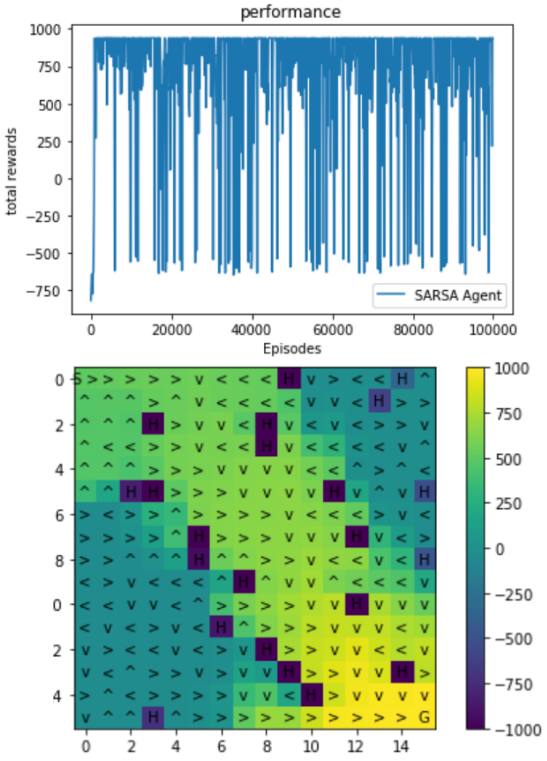
learning rate = 0.001, time = 328.003129482



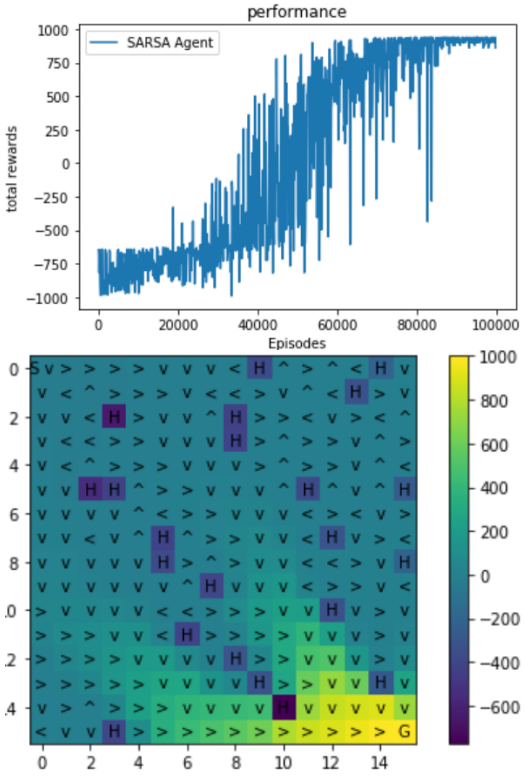
- **SARSA agent**

(epsilon = 0.3)

learning rate = 0.1, time = 224.5626928

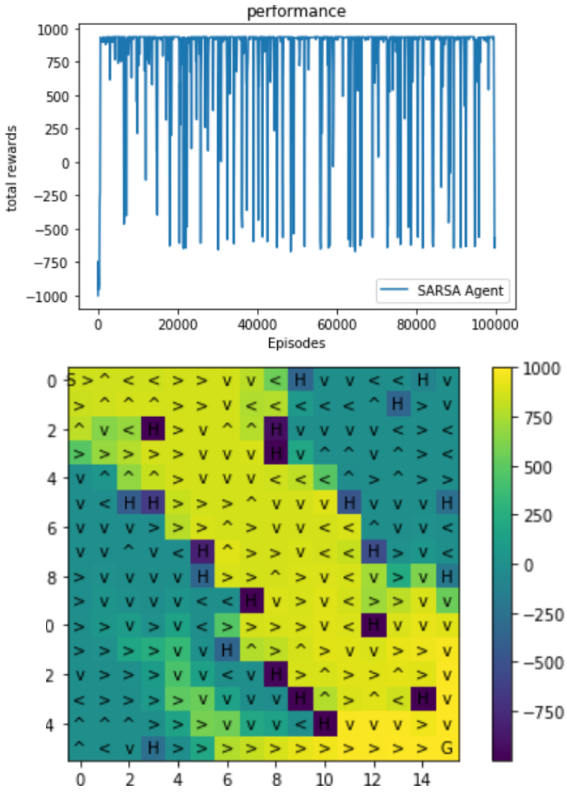


learning rate = 0.001, time = 514.662932157

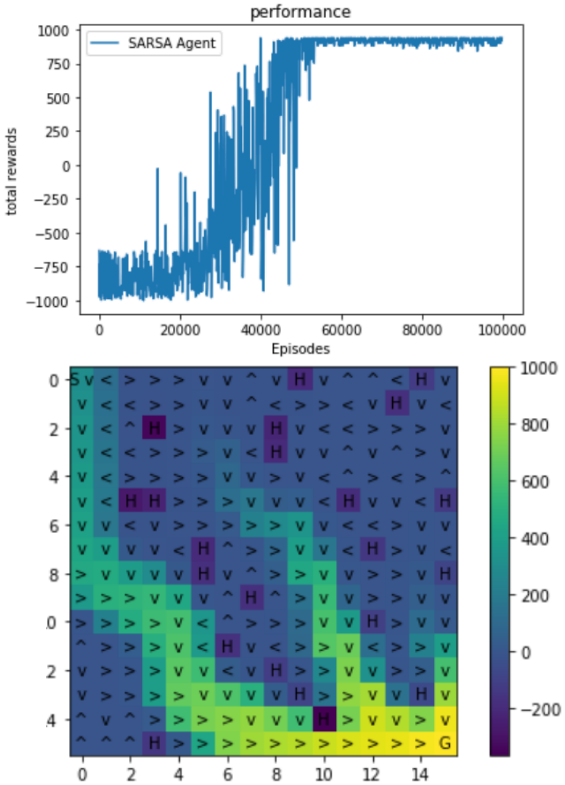


(epsilon = 0.05)

learning rate = 0.1, time = 160.877450942



learning rate = 0.001, time = 314.65030288

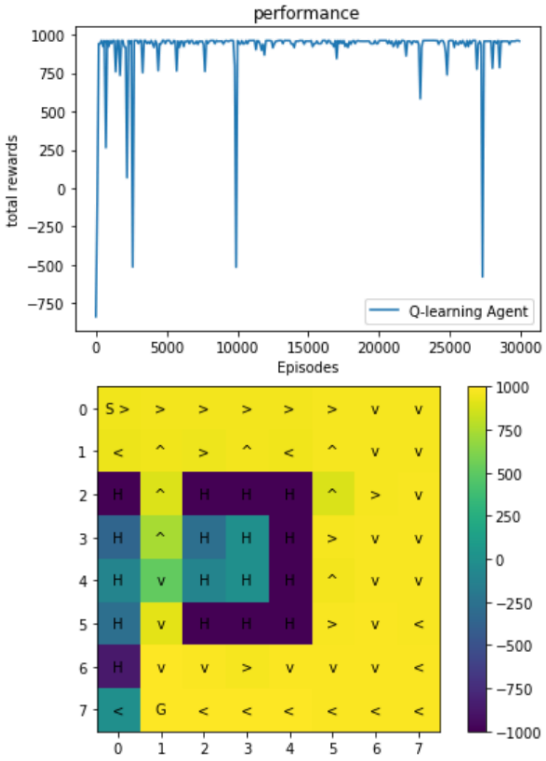


Map_Dangerous Hallway (learning episodes = 30,000, test_interval = 100)

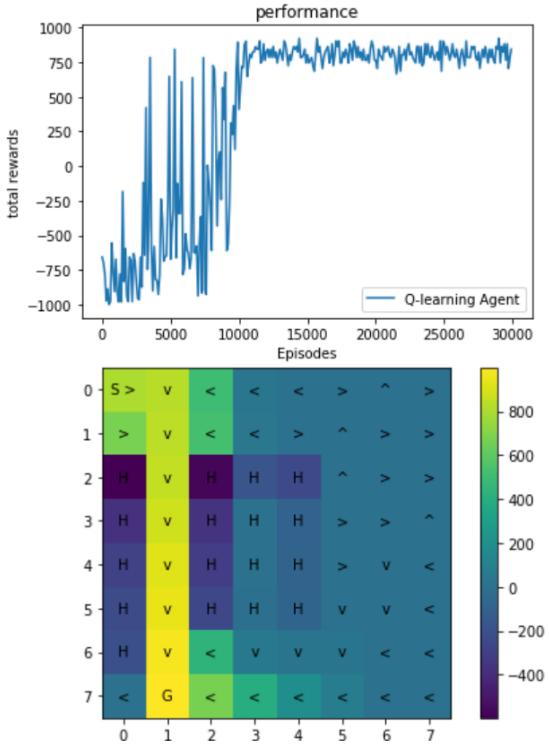
- **Q-learning agent**

(epsilon = 0.3)

learning rate = 0.1, time = 21.6749114

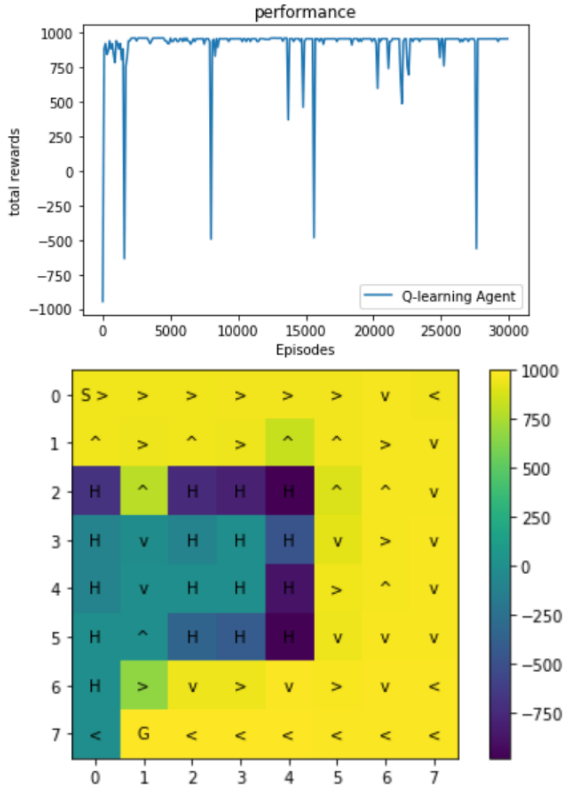


learning rate = 0.001, time = 50.8748414516

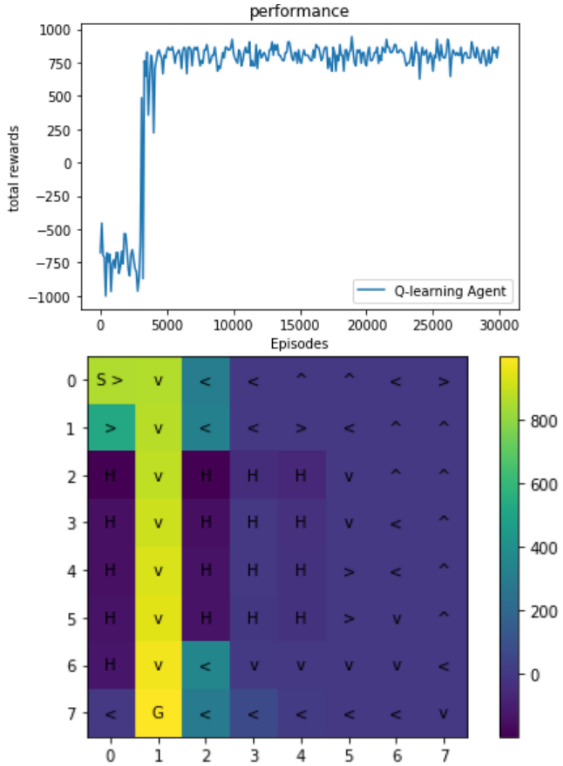


(epsilon = 0.05)

learning rate = 0.1, time = 19.5412466526



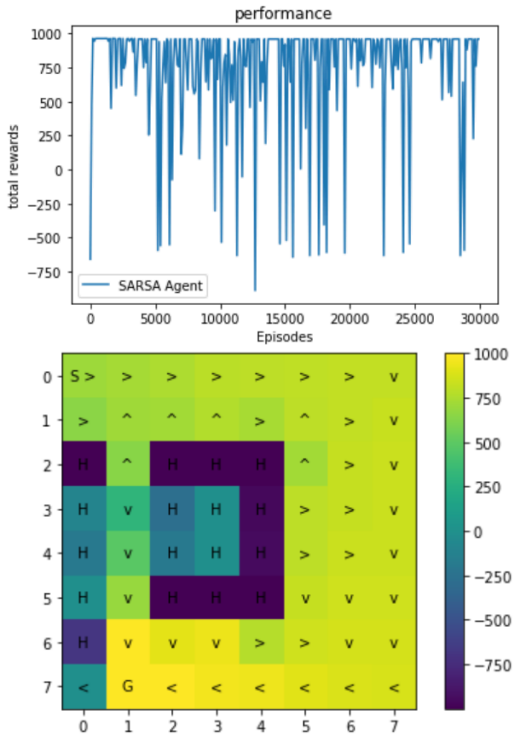
learning rate = 0.001, time = 25.672483205



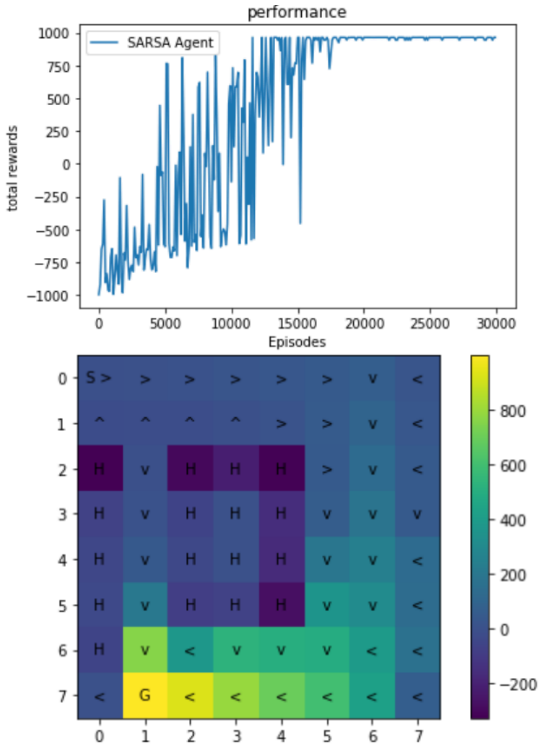
- **SARSA agent**

(epsilon = 0.3)

learning rate = 0.1, time = 54.5438578

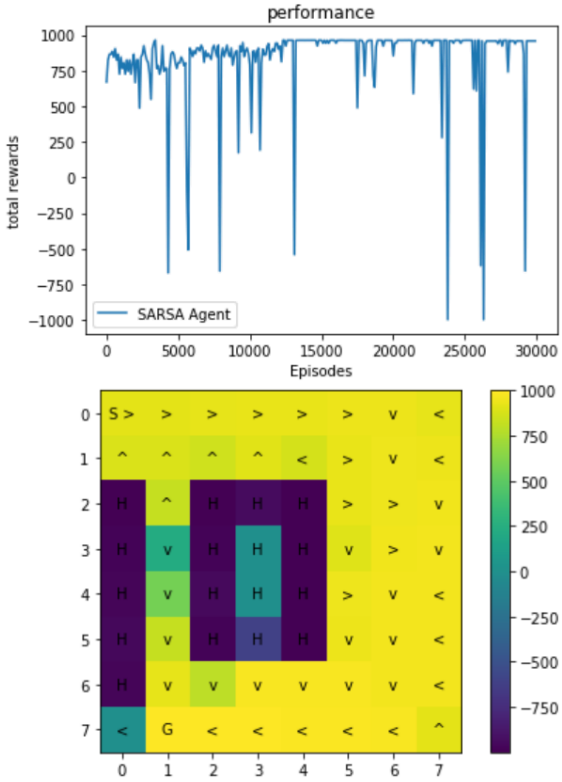


learning rate = 0.001, time = 79.8700001239

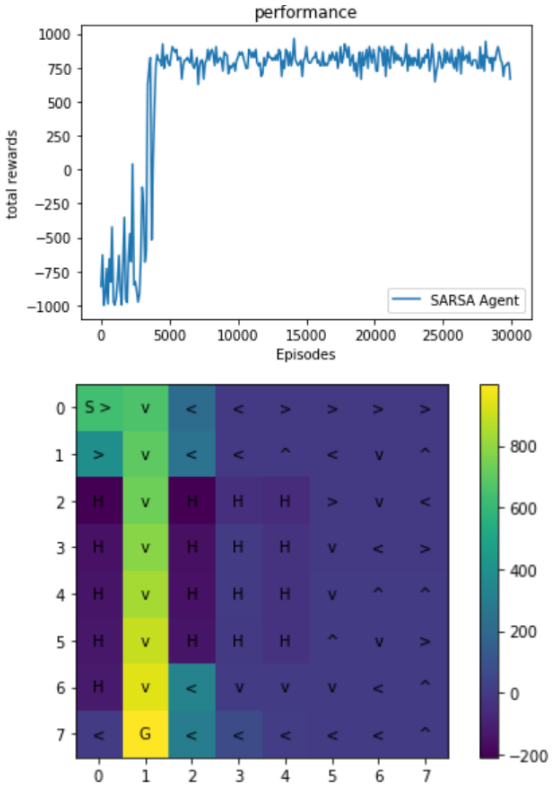


(epsilon = 0.05)

learning rate = 0.1, time = 24.341814041



learning rate = 0.001, time = 22.713646411



2. Did you observe differences for SARSA when using the two different learning rates? If there were significant differences, what were they and how can you explain them?

   There were significant differences when using two different learning rates. According to results showed above, small learning rate mostly need a longer learning time, and a bigger learning rate would tend to result in learning a suboptimal path and an overall faster learning time. Also, the coverage of high Q-value on the map was reduced when the learning rate gets small.

   Based on the algorithm, learning rate will have effect on reward updating. When having a small learning rate, reward updating will tend to be small, along with the difference between the goal and the estimated Q-value will also be small, which leads to the need of longer learning time, but also derives a better result. If the learning rate is big, the difference of the rewards after updating will be bigger, and will likely to cause entering a dead end or a sub optimal solution.

3. Repeat (2) for Q-Learning.

   Based on the results, two different learning rates do make a great differences on learning time, Q-value and policy map in Q-learning. As explained in (2), according to the algorithm, learning rate controls each step size of the difference between the estimated Q-value and the goal, which means that a large learning rate will result in a faster convergence.

4. Did you observe differences for SARSA when using different values of ϵ ? If there were significant differences, what were they and how do you explain them?

   Through our observation, the significantly differences between different values of ϵ is the policy and the convergence time. According to the explore/exploit policy, when it is given a bigger epsilon, it will have a bigger chance to explore the map. In contrast, when it is given a smaller ϵ, there won't be much of exploration, thus leads to a shorter path to the goal and also faster to convergence.

5. Repeat (4) for Q-Learning.

   Through our observation, the difference between different values of ϵ is also the the policy and the convergence time; however, for Q-learning, the difference isn't as obvious as SARSA. As explained in (4), with the explore/exploit policy, the chance of doing explore or exploit is depend on the value of ϵ. The reason of the subtle difference in Q-learning is that it only uses the explore/exploit policy for the first step, so it doesn't has great influence on the Q-learning agent.

6. For the map "Dangerous Hallway" did you observe differences in the policies learned by SARSA and Q-Learning for the two values of epsilon (there should be differences between Q-learning and SARSA for at least one value)? If you observed a difference, give your best explanation why Q-learning and SARSA found different solutions.

   The most significant difference between Q-learning and SARSA is when setting the ϵ to 0.3, the Q-learning always choose the shortest path to the goal; however, SARSA choose a safer path (goes around the side of map).

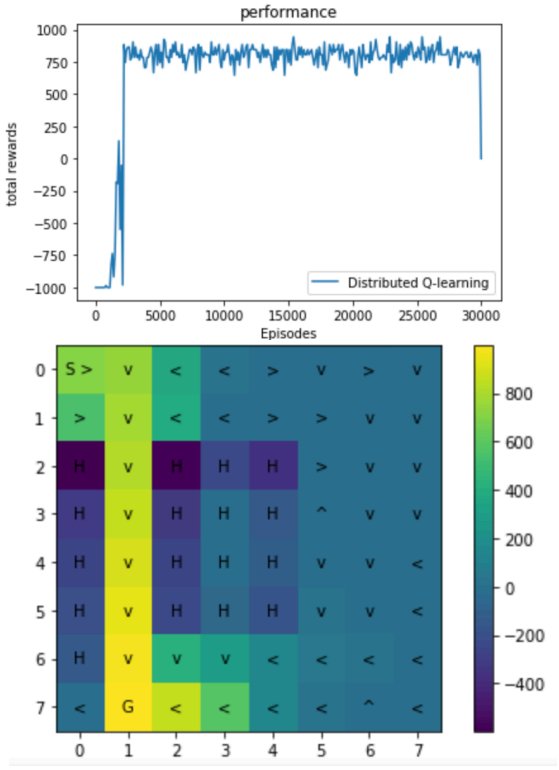
   According to both of the algorithms, when it comes to choosing the next step, Q-learning always goes for the maximum estimation using the greedy strategy, and SARSA uses the explore/exploit policy, allows it to have a chance to explore for safer path.

7. Show the value functions learned by the distributed methods for the best policies learned with ϵ equal to 0.3 and compare to those of the single-core method. Run the algorithm for the recommended number of episodes for each of the maps. Did the approaches produce similar results?

   (collector workers, evaluator workers) = **(2,4)**

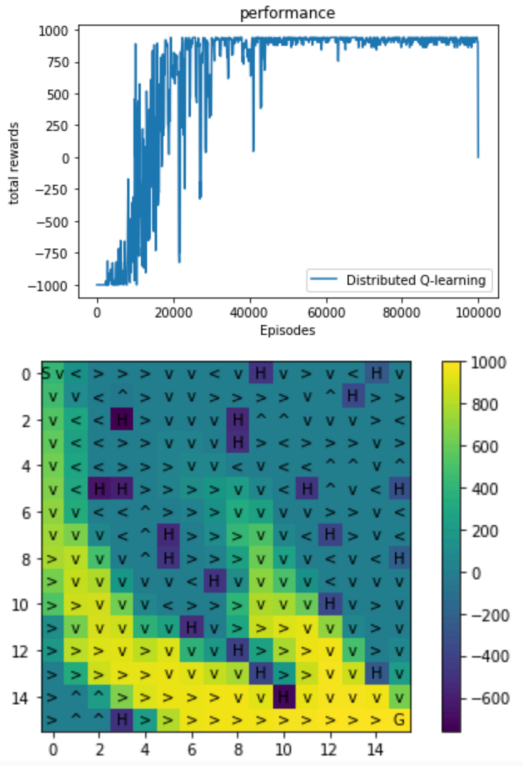
   Map_Dangerous Hallway

   time: 9.520294666290283



   Map_16x16

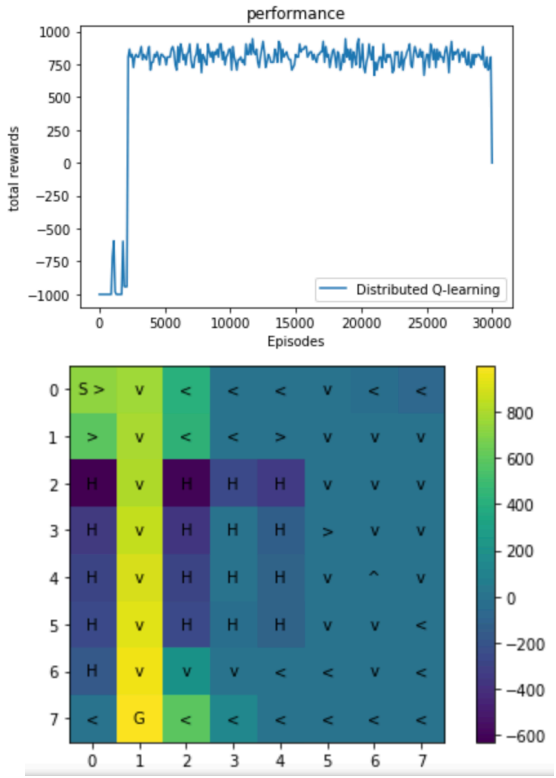
   time: 59.523006200790405



(collector workers, evaluator workers) = **(4,4)**

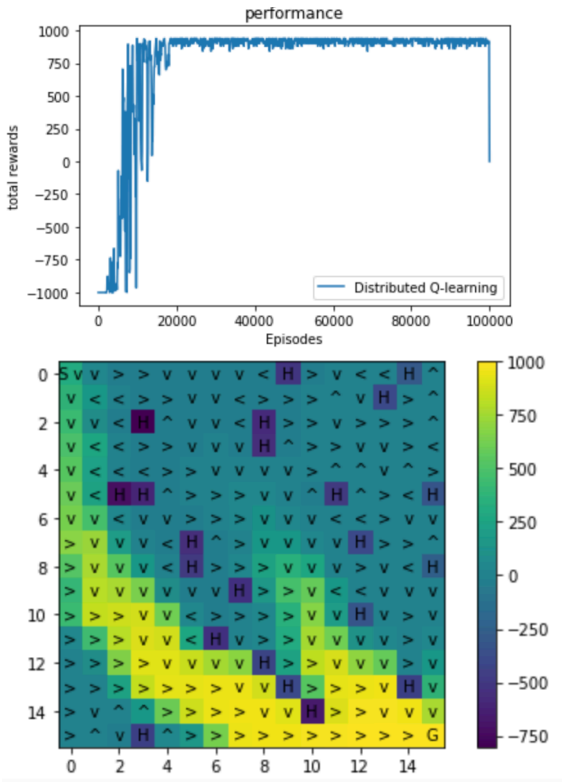
Map_Dangerous Hallway

time: 4.993547677993774



Map_16x16

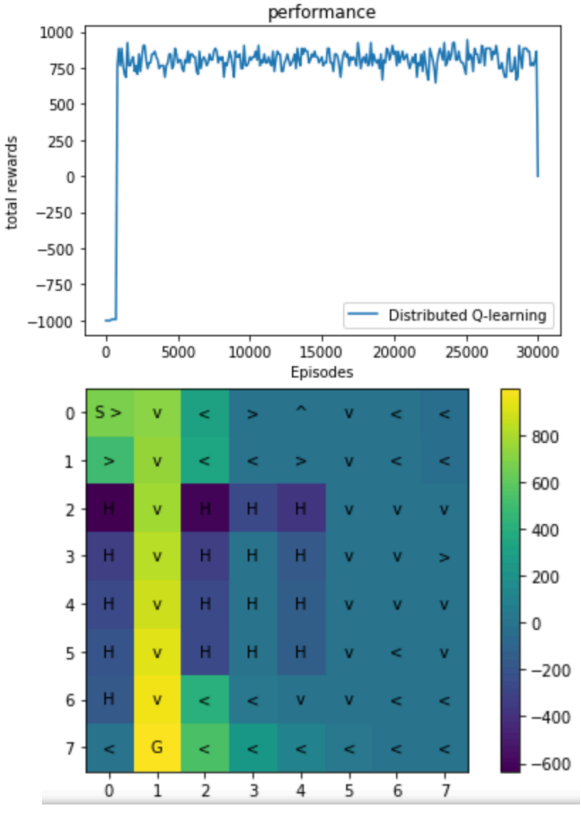
time: 38.3487843193054



(collector workers, evaluator workers) = **(8,4)**

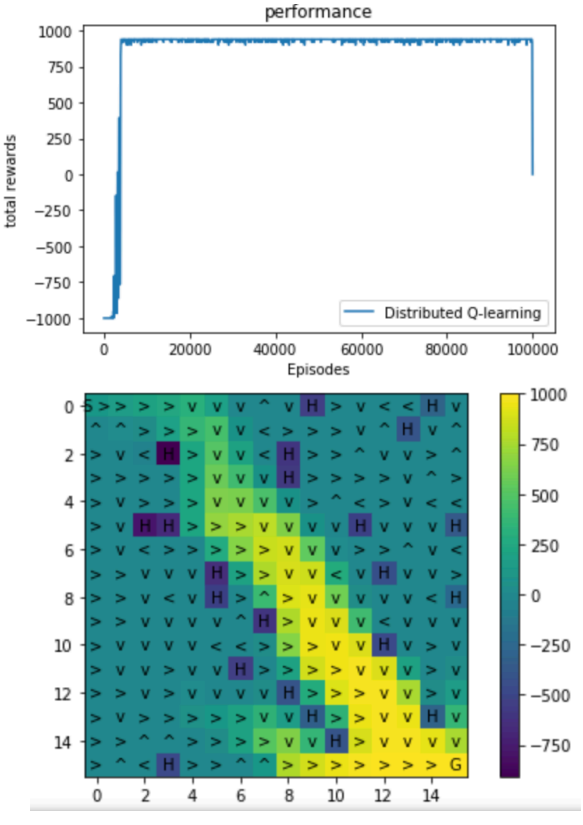
Map_Dangerous Hallway

time: 3.874885082244873



Map_16x16

time: 32.155010903864441



The value function and policy are similar with the single-core method. The only differences in the 16*16 map policy. However, the performances are significant improved. The distributed methods converge much faster than the single-core method.

8. Provide and compare the timing results for the single-core and distributed experiments, including the time to do the evaluations during learning. Describe the trends you observe as the number of workers increases.
According to the time results, the time significantly decreased when changing single-core to two core. Then the more core we used, the faster it became. The time difference between (4,4) and (8,4) doesn't have significant increase like it does between single-core and two core, this may because of the server is handling too much works.

	Map_DH	Map_16x16
Single-core	50.8748414516	461.642726182
Distributed(2,4)	9.520294666290280	59.523006200790400
Distributed(4,4)	4.993547677993770	38.3487843193054
Distributed(8,4)	3.874885082244870	32.155010903864400

9. Provide and compare the timing results for the single-core and distributed experiments with the evaluation procedure turned off. That is, here you only want to provide timing results for the learning process without any interleaved evaluation. Compare the results with (8).
Without the evaluation procedure, all the experiences including single core became much faster than before.

	Map_DH	Map_16x16
Single-core	51.8304777145385	135.898846149444
Distributed(2,4)	6.81975793838501	51.4750328063964
Distributed(4,4)	4.59111452102661	27.9862079620361
Distributed(8,4)	2.29641485214233	18.8017807006835