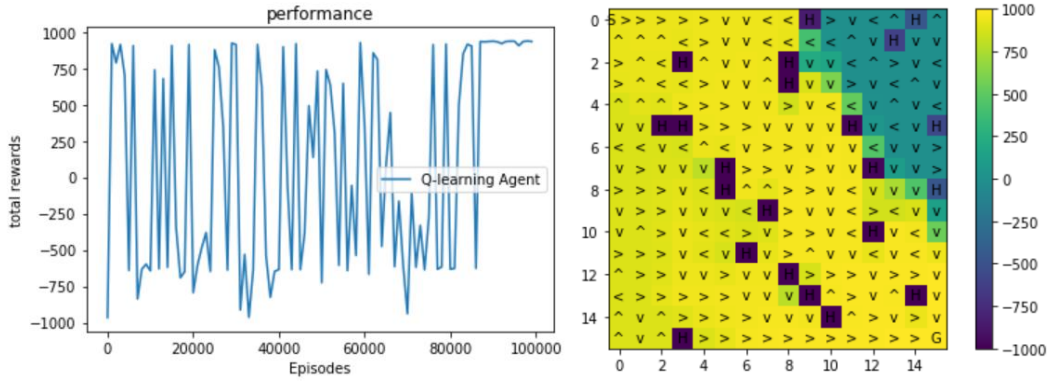
CS533 HW3

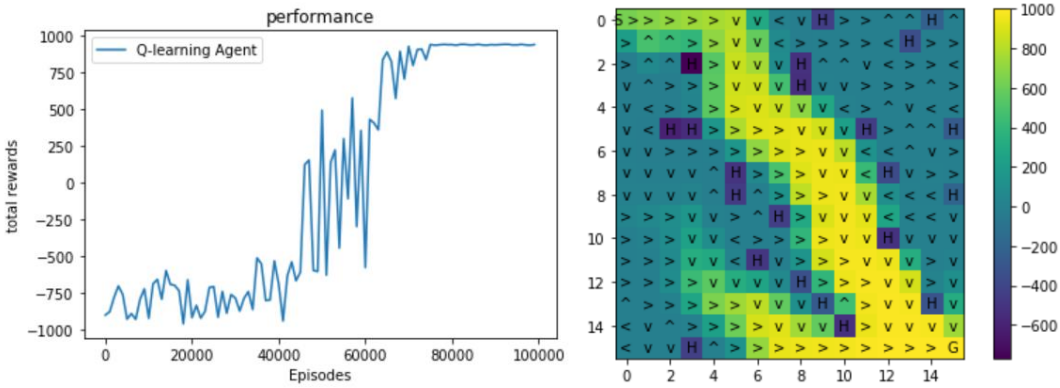
1. Provide the learning curves for the above experiments. Clearly label the curves by the parameters used.
   MAP_ 16X16 (LEARNING_EPISODES=100,000, TEST_INTERVAL=1000)

Q-LEARNING

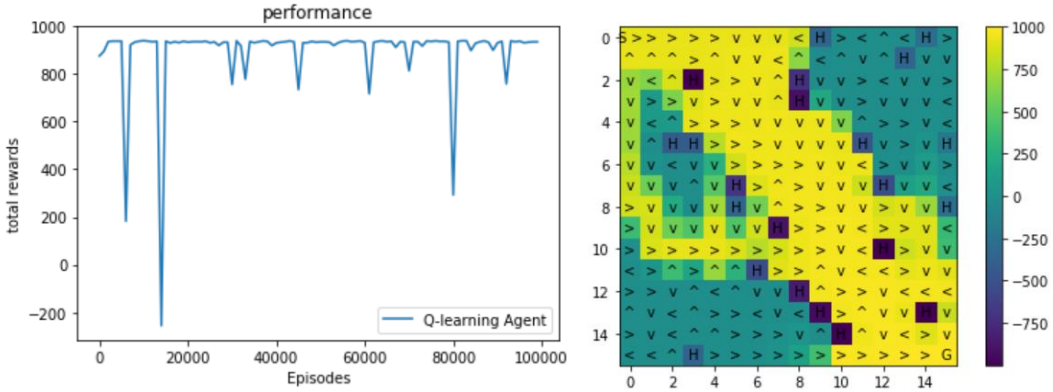
EPISLON=0.3, LEARNING_RATE=0.1, EXE_TIME = 494.2039420604706



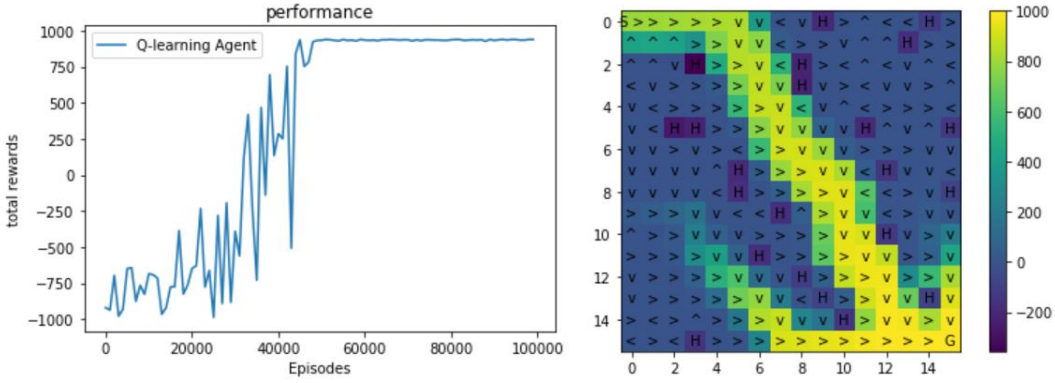
EPISLON=0.3, LEARNING_RATE=0.001, EXE_TIME = 618.7666273117065, 461.9668414592743



EPISLON=0.05, LEARNING_RATE=0.1, EXE_TIME = 113.06760430335999



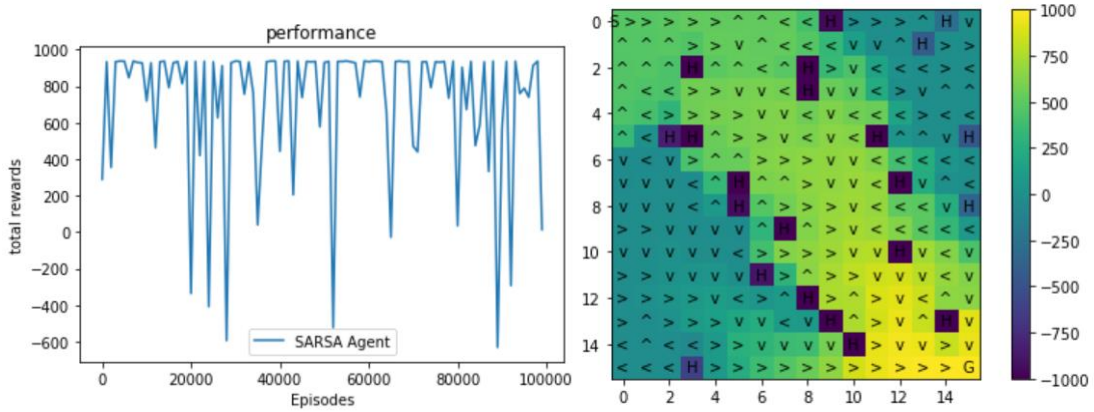
EPISLON=0.05, LEARNING_RATE=0.001, EXE_TIME = 326.60723757743835



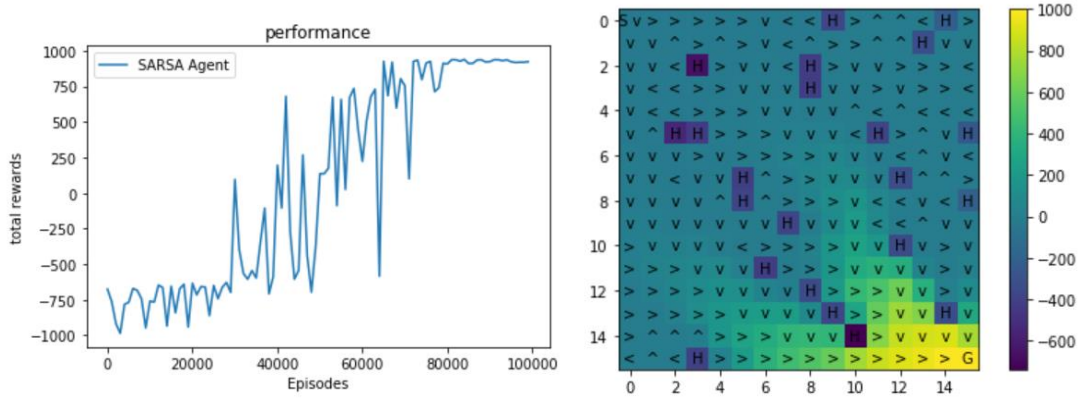
CS533 HW3

SARSA

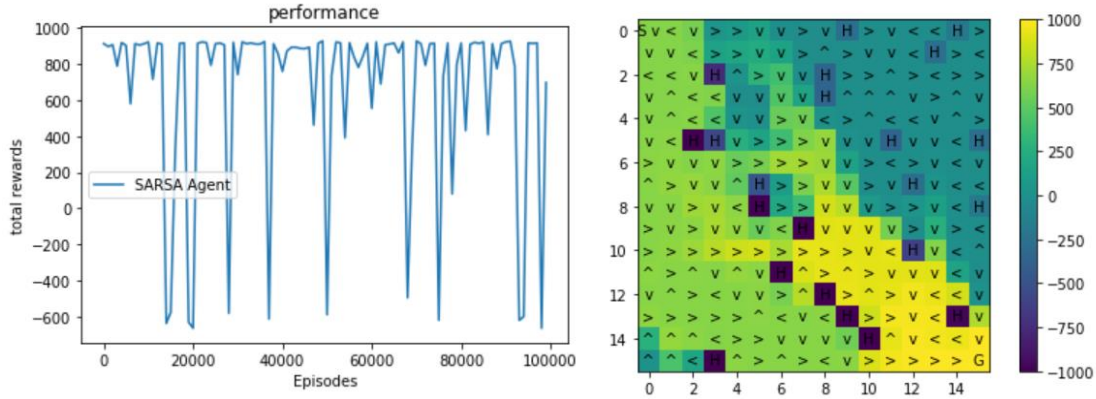
EPISLON=0.3, LEARNING_RATE=0.1, EXE_TIME = 256.4520471096039



EPISLON=0.3, LEARNING_RATE=0.001, EXE_TIME = 556.9661309719086

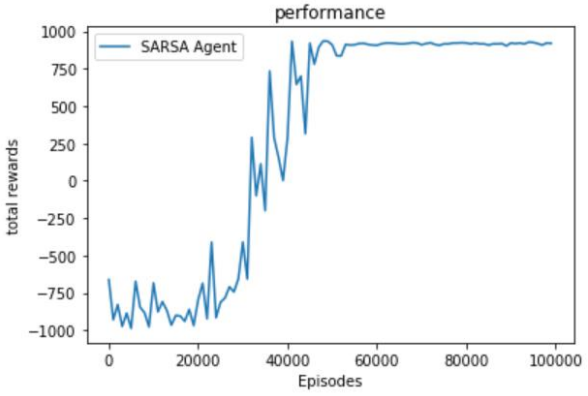


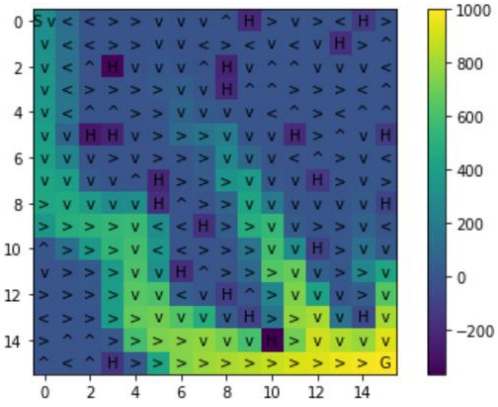
EPISLON=0.05, LEARNING_RATE=0.1, EXE_TIME = 276.6897075176239



EPISLON=0.05, LEARNING_RATE=0.001, EXE_TIME = 297.53178119659424

CS533 HW3


performance
SARSA Agent
total rewards
Episodes
1000
750
500
250
0
−250
−500
−750
−1000
0
20000
40000
60000
80000
100000

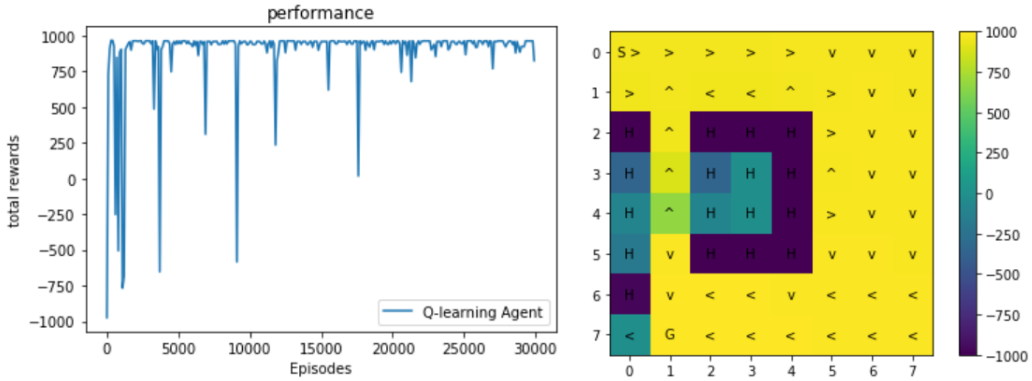

0
2
4
6
8
10
12
14
1000
800
600
400
200
0
−200

CS533 HW3

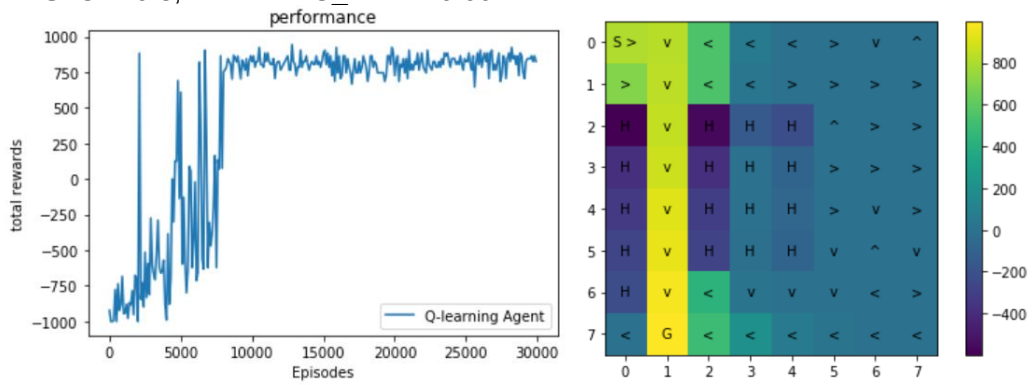
MAP_ Dangerous Hallway (LEARNING_EPISODES=30,000, TEST_INTERVAL=100)

Q-LEARNING

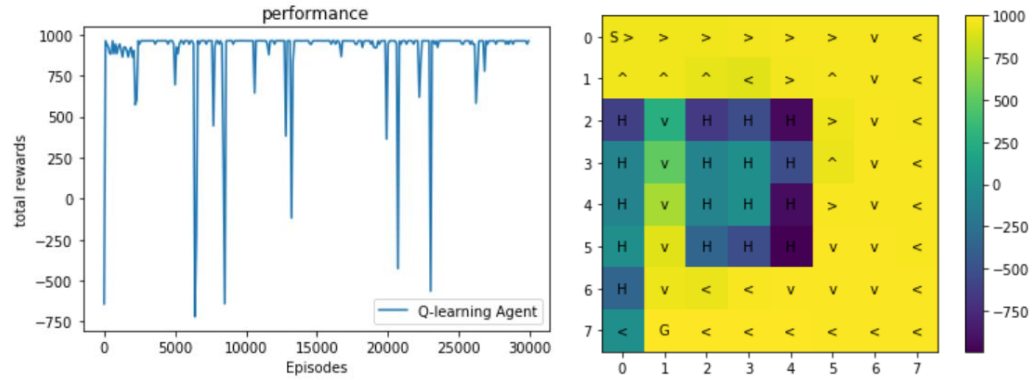
EPISLON=0.3, LEARNING_RATE=0.1



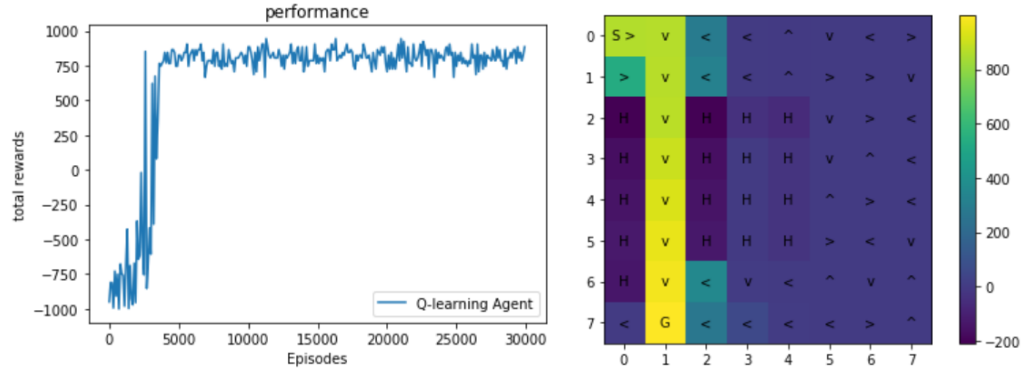
EPISLON=0.3, LEARNING_RATE=0.001



EPISLON=0.05, LEARNING_RATE=0.1

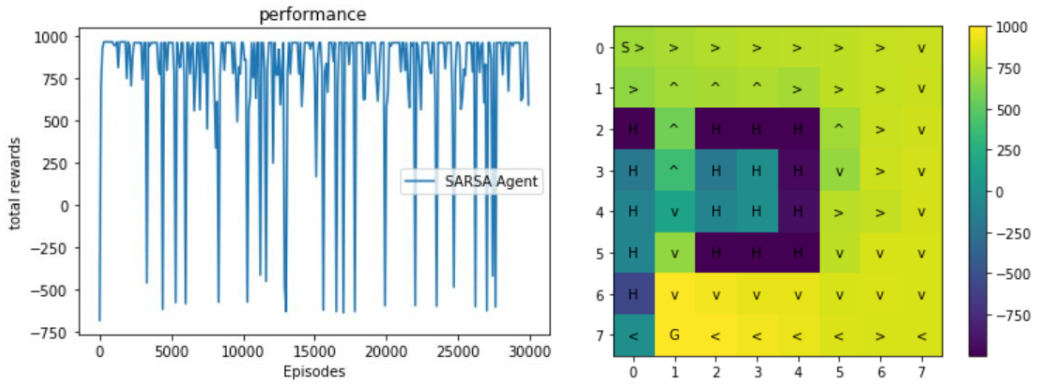


EPISLON=0.05, LEARNING_RATE=0.001

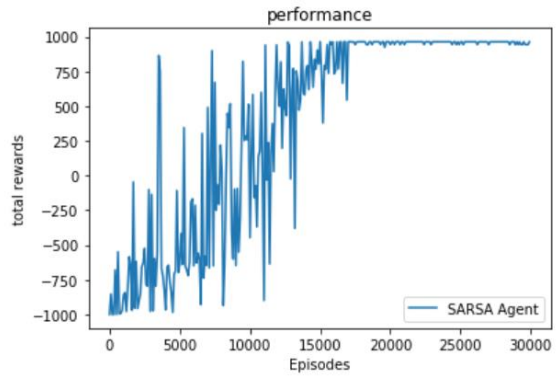


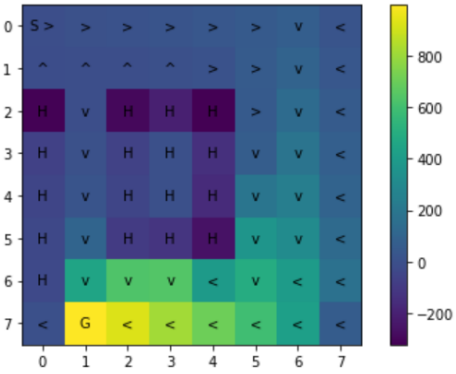
SARSA

EPISLON=0.3, LEARNING_RATE=0.1

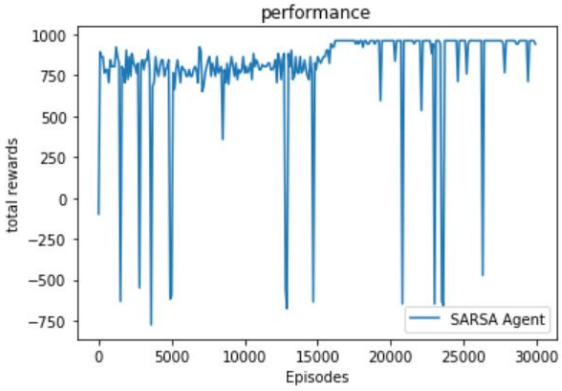


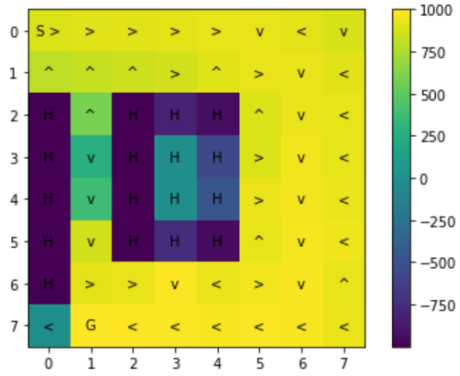
EPISLON=0.3, LEARNING_RATE=0.001





EPISLON=0.05, LEARNING_RATE=0.1





EPISLON=0.05, LEARNING_RATE=0.001

CS533 HW3

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?

   According to the TD update for transition from s to s', $the\ learning\ rate,\ \alpha_n = \frac{1}{n}\ is\ a\ valid\ choice$ because of convergence.
   (We must satisfy $\sum_{n=1}^{\infty} \alpha_n = \infty$ and $\sum_{n=1}^{\infty} \alpha_n^2 < \infty$.)
   The relationship between the learning rate and the sample size is inverse. As the learning rate decreases, the sample size increases, and the convergence can be guarantees.
   Also, it could be related to the accuracy of the learning and evaluation. As you can see in the figures, as the learning rate decreases, each cell is more distinguishable. It reflects the convergence and the accuracy of the result.

3. Repeat (2) for Q-Learning.
   As explained above, question 2, the learning rate affects its convergence and the goal finding in Q-Learning as well.

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

   With the higher Epsilon value, the agent will explore more than exploit.
   The different result from SARSA ALG (EP 0.3 and 0.05) show that the agent with the higher EP value can find more state to go than the lower EP value.

5. Repeat (4) for Q-Learning.
   As you can see the figures above, Q-Learning ALG EP 0.3 and 0.05 show the very similar result as SARSA ALG. The effect of EP change provides the same results in SARSA and Q-Learning.

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 for why Q-learning and SARSA found different solutions.

   Q-learning can find the shortest path to the “G” point.
   SARSA will find the safer path to the “G” point.

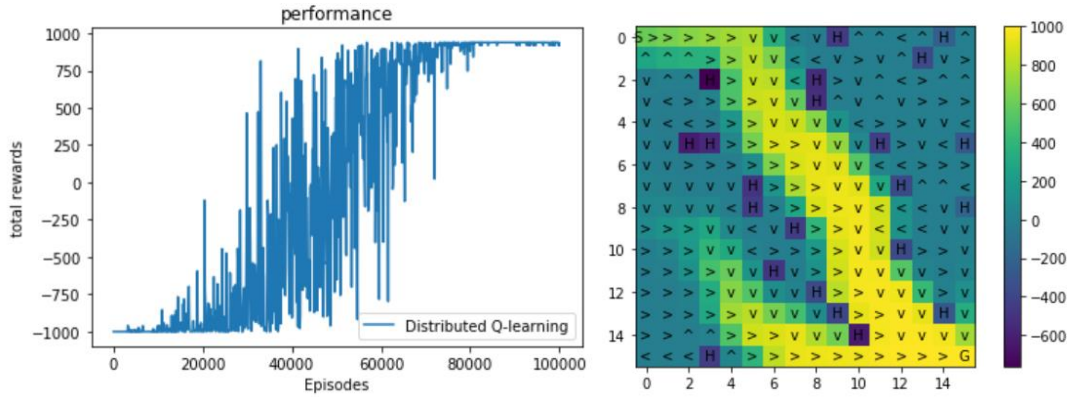
   Because the ALG of Q-learning find a next Action by finding maximize argument A of Q function and SARSA find a next action by using ϵ -greedy policy.

7. Show the value functions learned by the distributed methods for the best policies learned with $\epsilon\epsilon$ 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?

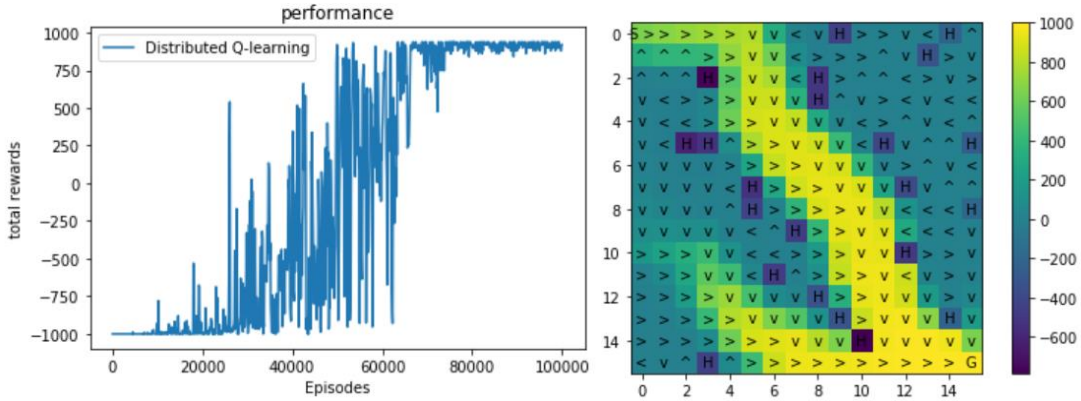
MAP_ 16

⇨ LEARNING_EPISODES=100,000, TEST_INTERVAL=100, BATCH_SIZE=10
EPSILON=0.03, LEARNING_RATE=0.01

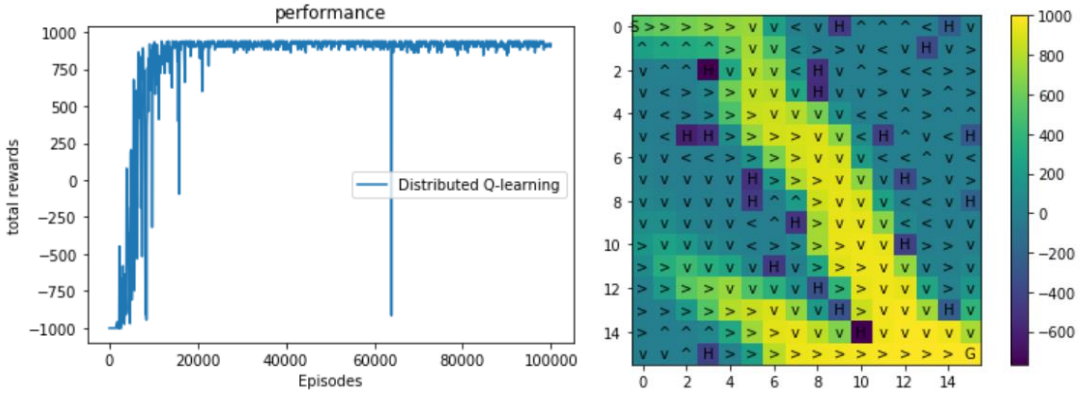
WORKER = (2, 4)



WORKER = (4, 4)



WORKER = (8,4)

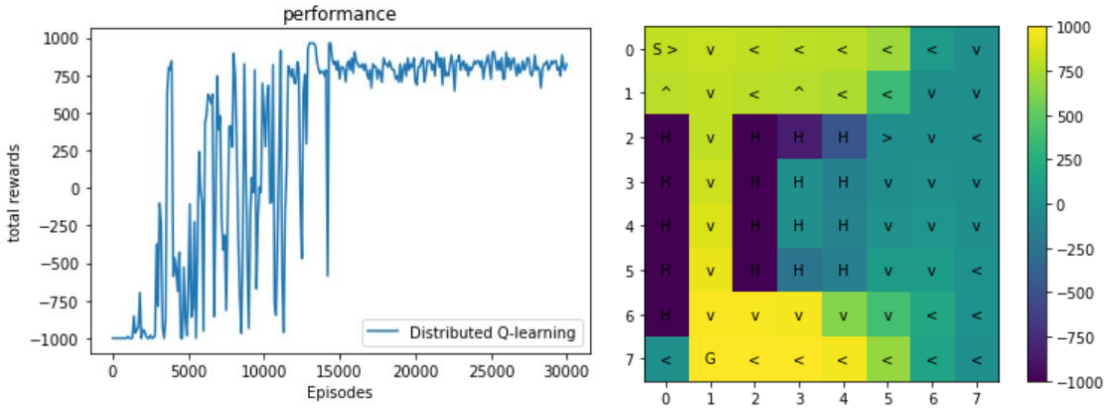


CS533 HW3

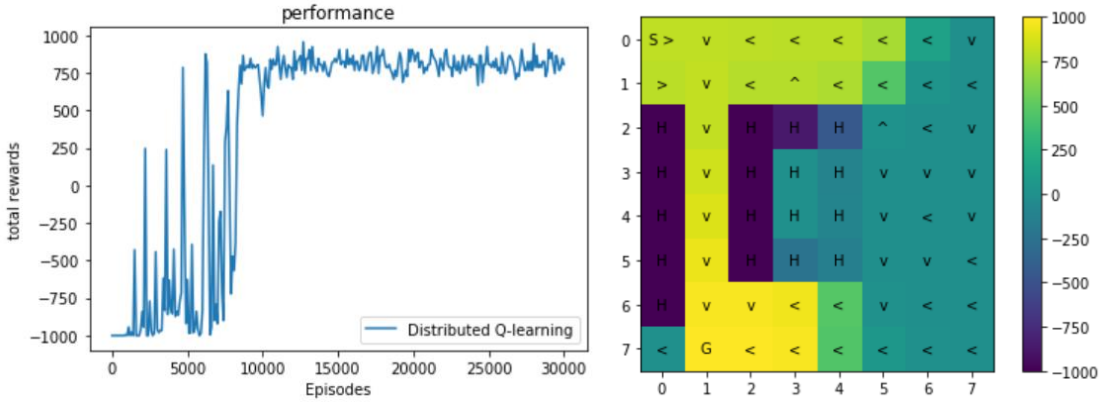
MAP_ Dangerous Hallway

⇨ LEARNING_EPISODES=30,000, TEST_INTERVAL=100, BATCH_SIZE=10, EPSILON=0.3, LEARNING_RATE=0.01

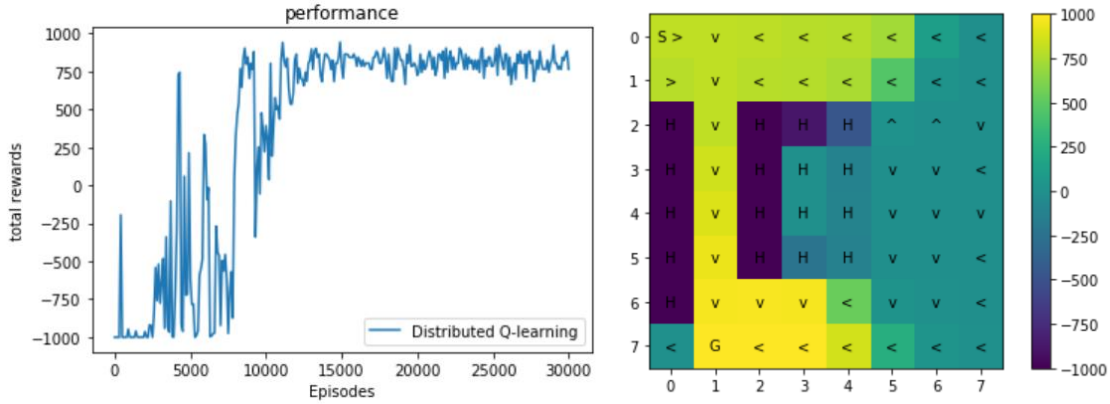
WORKER = (2, 4)

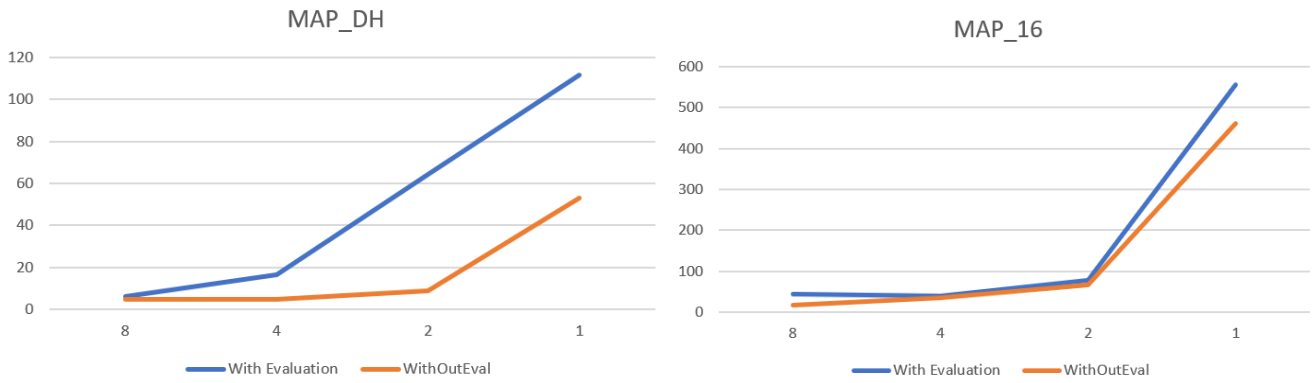


WORKER = (4, 4)



WORKER = (8,4)





* Single-Core => row number = 1
* Distributed => row number = workers number

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.

   2 workers have improved the total running time by 2 times faster than the single-core in the MAP_DH. 4 and 8 of workers have improved the total running time by 4 times faster than the single-core in the MAP_DH.

   In the MAP_16, the running time of single-core is way slower than the expectation. It might be due to the busy server. So, the running time of the distributed experiment has been improved more than 4 times.

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, of course, all distributed experiments get the faster executing time than the single-core one because multiple workers collects the experience as the learning result.