Reinforcement Learning: Assignment #5

Due on Sunday, Mai 14, 2023

Well done!

9

p.s. you're already reached 50%

congrats ☺

Group 4 - Abu El Komboz, Tareq 3405686 | Jain, Likhit 3678905 | Wurm, Marcel 3695946

Task 1

Random Walk (2P)

(a) Recall the Random walk example presented in the lecture (lecture 5 slide 12). From the results shown in the right graph above (estimated value) it appears that the first episode results in a change in only V(A). What does this tell you about what happened on the first episode? Why was only the estimate for this one state changed? By exactly how much was it changed (assuming α = 0.1)? Support your answers by computing the TD-update.

In our given setting, only the estimate for the last state before the terminal state is updated in the first episode. This outcome tells us that the first episode ended up in the left terminal state. For all other states along the episode, which didnt lead to a terminal state, the estimate didn't change because
$V(s) \leftarrow V(s) + \alpha[R + \gamma V(s') - V(s)] = V(s) + \alpha[0 + 1 \cdot 0.5 - 0.5] = V(s)$.

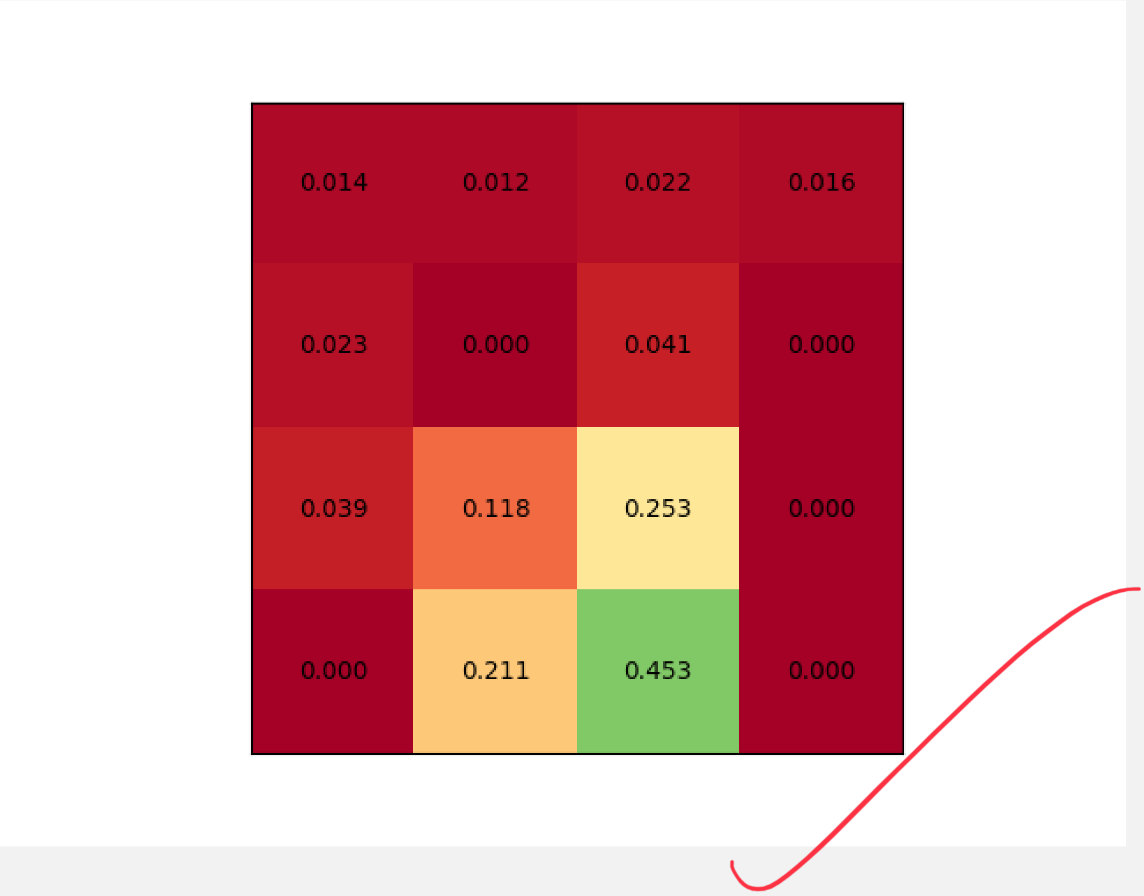
TD-Update for s_A:
$V(s_A) \leftarrow V(s_A) + \alpha[R + \gamma V(s_{\text{left-terminal}}) - V(s_A)] = 0.5 + 0.1[0 + 1 \cdot 0 - 0.5] = 0.45$

Task 2

Sarsa and Q-learning on the FrozenLake (8P)

(a) Implement Sarsa and obtain and plot the state-value function, action-value function, and policy for the FrozenLake environment. Plot the average episode length as training continues. (3P)

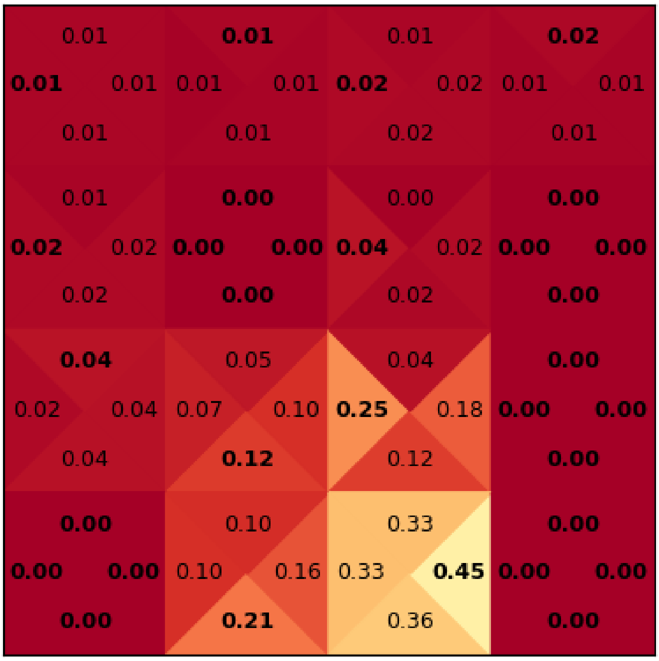
State-Value-Function:



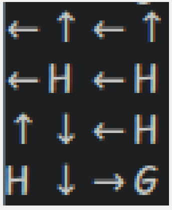
0.014	0.012	0.022	0.016
0.023	0.000	0.041	0.000
0.039	0.118	0.253	0.000
0.000	0.211	0.453	0.000

Task 2 continued on next page...

Action-Value-Function:

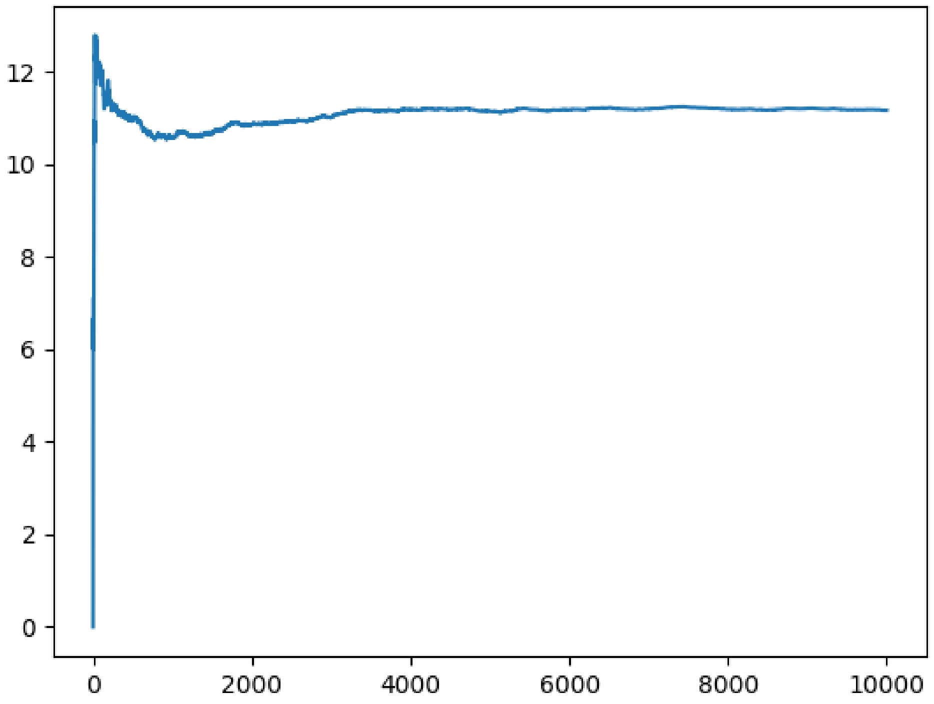


Policy:



```
← ↑ ← ↑
← H ← H
↑ ↓ ← H
H ↓ → G
```

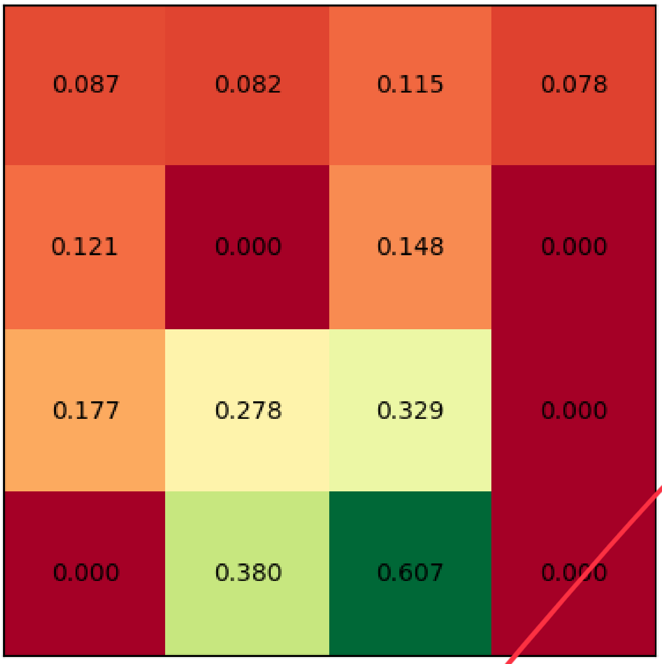
Average episode length:



Task 2 continued on next page...

(b) Implement Q-learning and obtain and plot the optimal state-value function, action-value function, and policy for FrozenLake. What can you say about performance during training in comparison to the performance of the optimal policy? (3P)

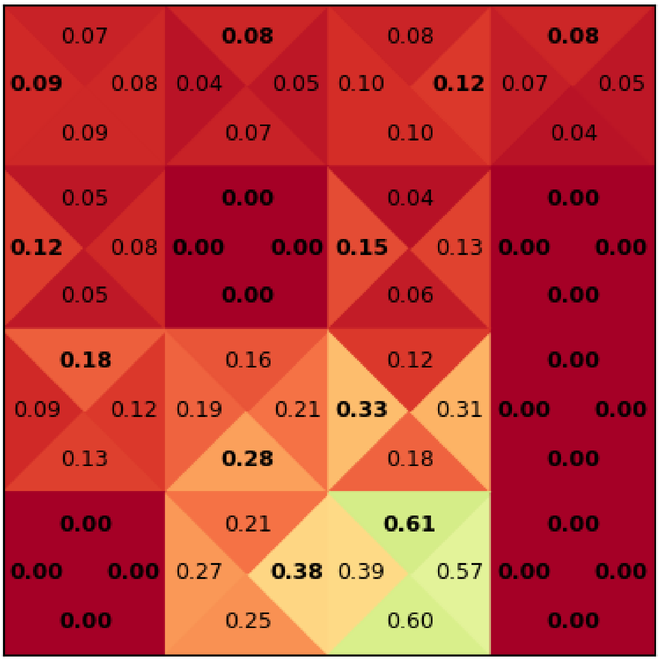
State-Value-Function:



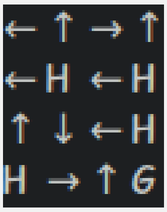
0.087	0.082	0.115	0.078
0.121	0.000	0.148	0.000
0.177	0.278	0.329	0.000
0.000	0.380	0.607	0.000

Task 2 continued on next page...

Action-Value-Function:



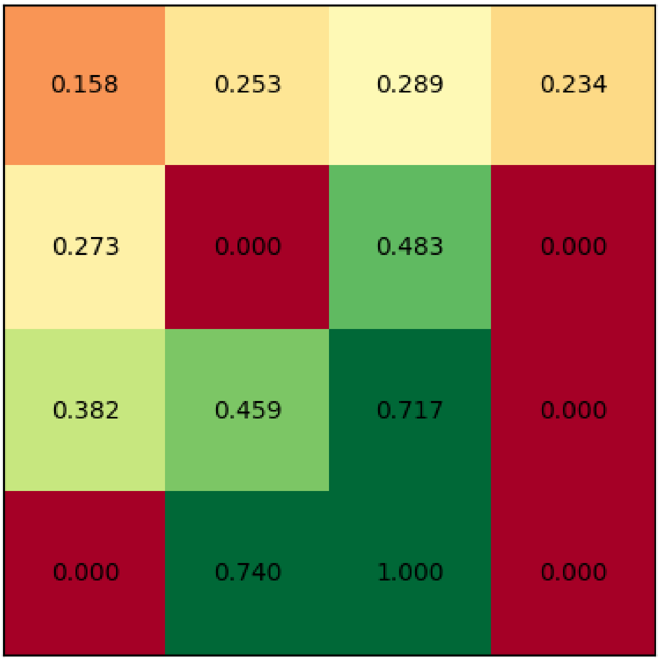
Policy:



```
← ↑ → ↑
← H ← H
↑ ↓ ← H
H → ↑ G
```

(c) Explore how your results for a) and b) change if you switch to the non-slippery version (i.e. deterministic environment). (1P)

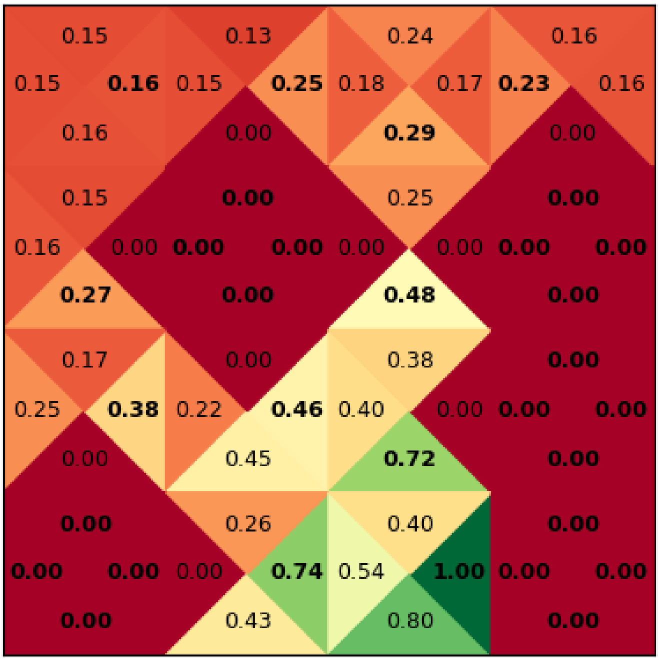
SARSA State-Value-Function:



0.158	0.253	0.289	0.234
0.273	0.000	0.483	0.000
0.382	0.459	0.717	0.000
0.000	0.740	1.000	0.000

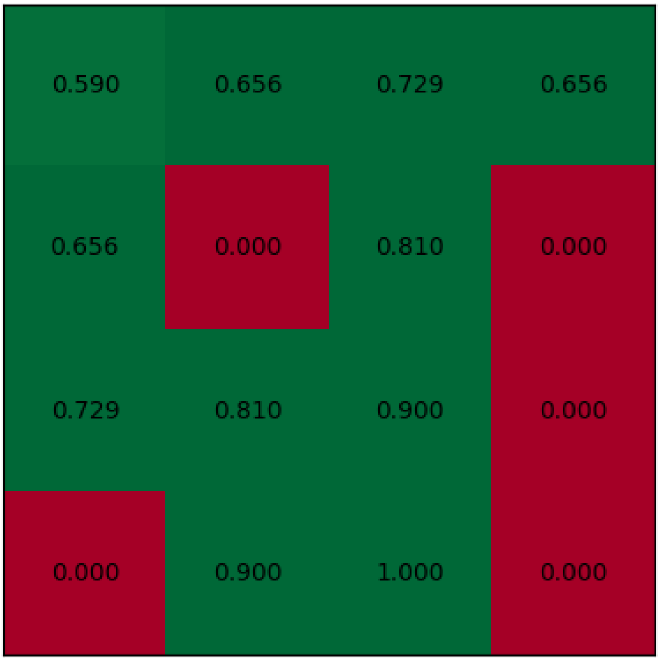
Task 2 continued on next page...

SARSA Action-Value-Function:



Task 2 continued on next page. . .

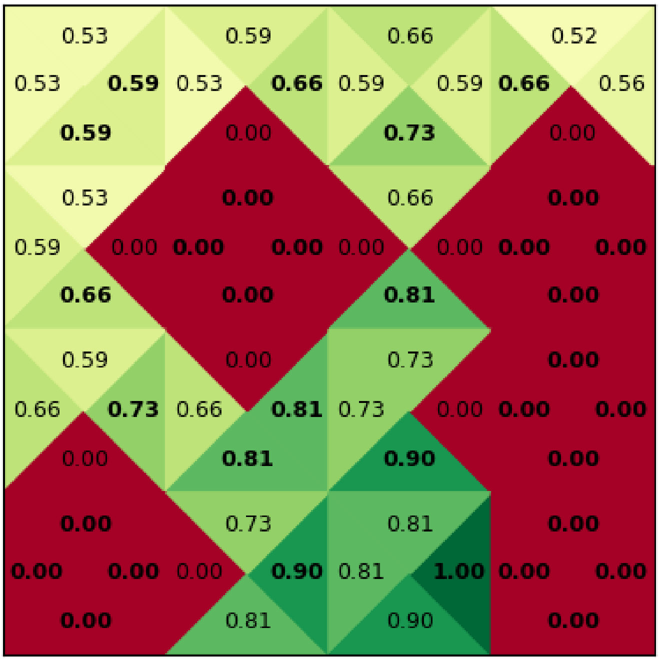
Q-Learning State-Value-Function:



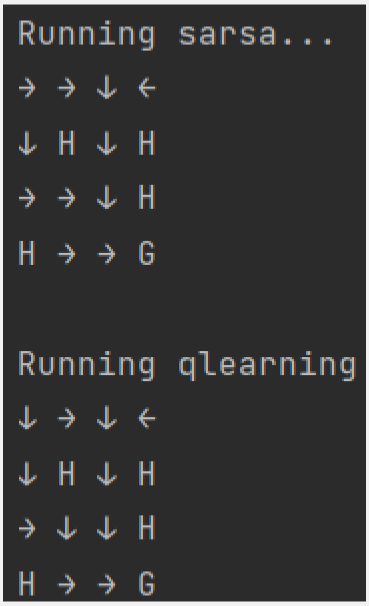
0.590	0.656	0.729	0.656
0.656	0.000	0.810	0.000
0.729	0.810	0.900	0.000
0.000	0.900	1.000	0.000

Task 2 continued on next page. . .

Q-Learning Action-Value-Function:



Policies:



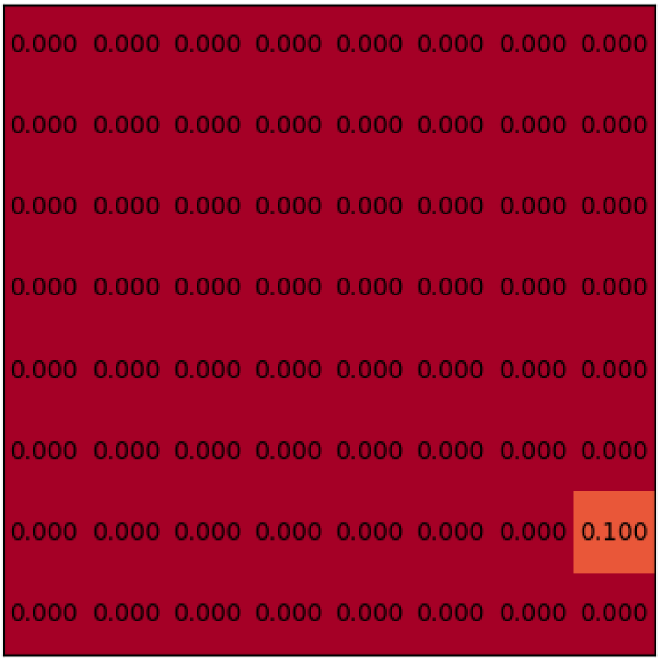
```
Running sarsa...
→ → ↓ ←
↓ H ↓ H
→ → ↓ H
H → → G

Running qlearning
↓ → ↓ ←
↓ H ↓ H
→ ↓ ↓ H
H → → G
```

Task 2 continued on next page...

(d) Rerun your code for the larger FrozenLake environment. (1P)

SARSA State-Value-Function:

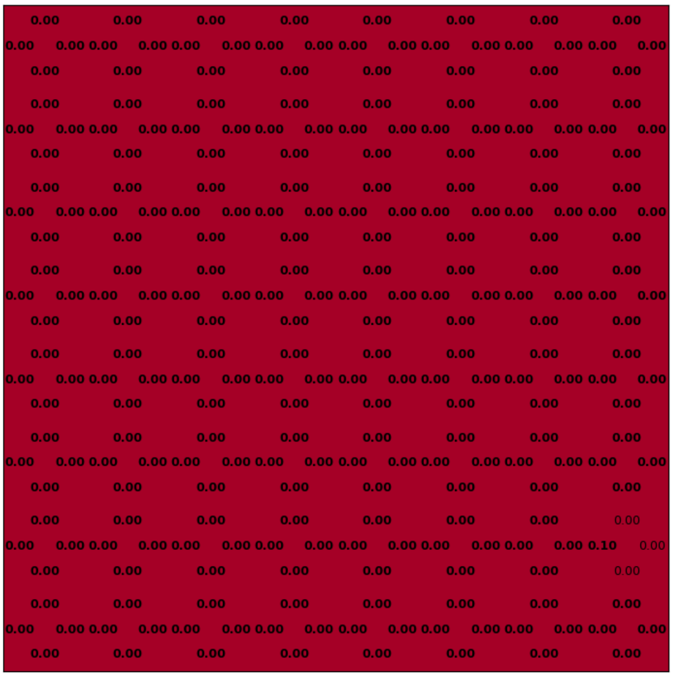


0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.100
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

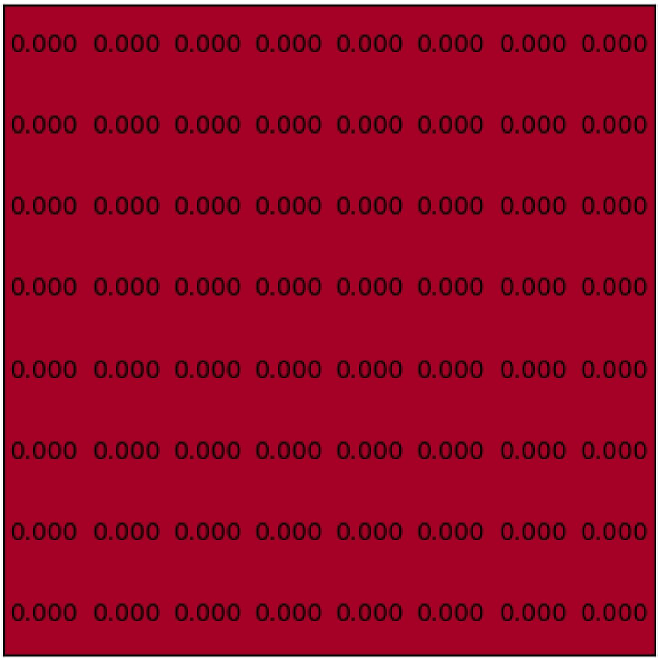
I don't think so . . .

-1

SARSA Action-Value-Function:

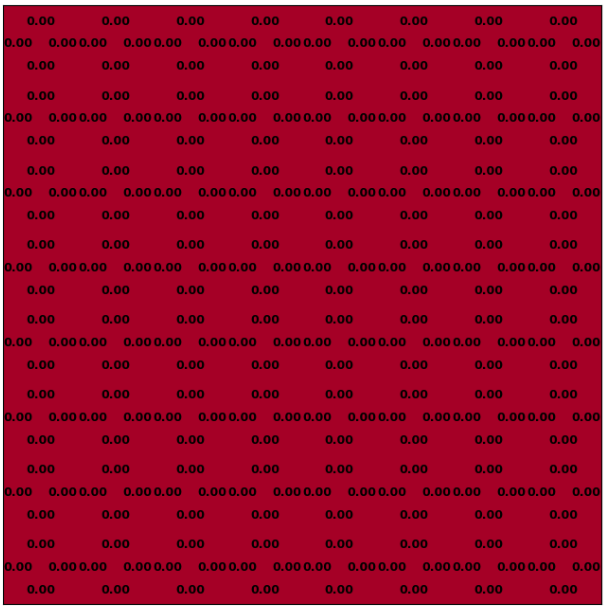


Q-Learning State-Value-Function:



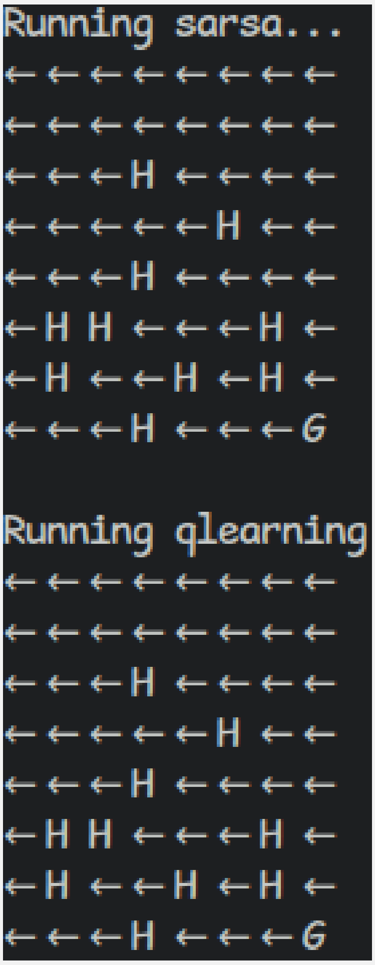
Task 2 continued on next page. . .

Q-Learning Action-Value-Function:



Task 2 continued on next page. . .

Policies:

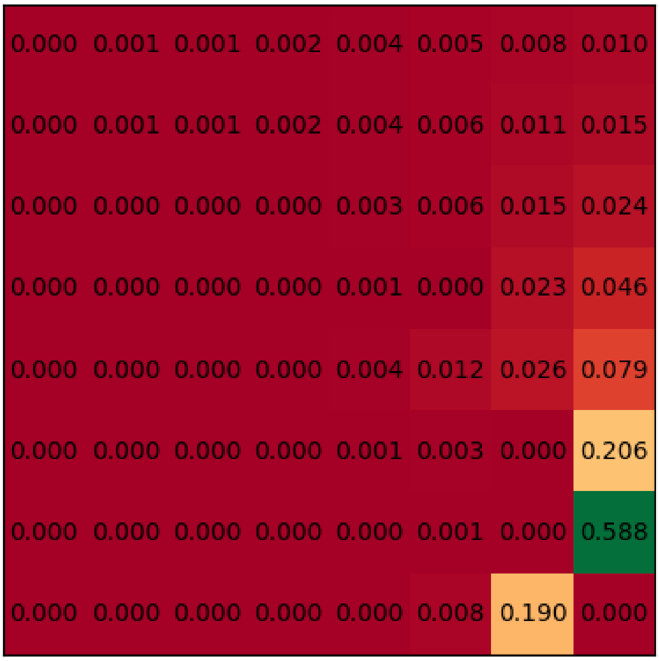


```
Running sarsa...
←←←←←←←←
←←←←←←←←
←←←H←←←←
←←←←←H←←
←←←H←←←←
←HH←←←H←
←H←←H←H←
←←←H←←←G

Running qlearning
←←←←←←←←
←←←←←←←←
←←←H←←←←
←←←←←H←←
←←←H←←←←
←HH←←←H←
←H←←H←H←
←←←H←←←G
```

Overall the environment size seems to be too big for our current algorithms and number of episodes. Thus the goal is never actually reached and no reward can be accumulated (Although we got lucky once with the sarsa algorithm and reached the goal).
When running with significantly more iterations (10.000 instead of 1.000), the following state-value-functions are produced which indicates some progress but still not enough to make the policy usable (And the required time was obviously also significantly more).

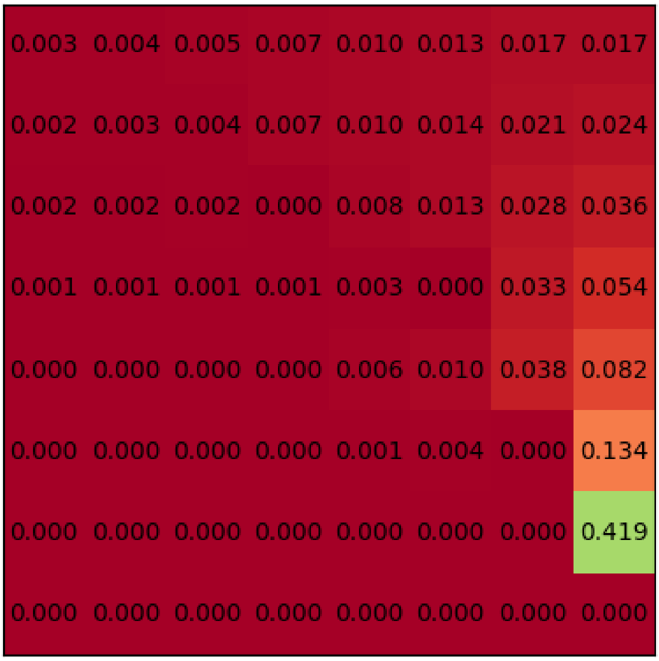
SARSA State-Value-Function with more iterations:



0.000	0.001	0.001	0.002	0.004	0.005	0.008	0.010
0.000	0.001	0.001	0.002	0.004	0.006	0.011	0.015
0.000	0.000	0.000	0.000	0.003	0.006	0.015	0.024
0.000	0.000	0.000	0.000	0.001	0.000	0.023	0.046
0.000	0.000	0.000	0.000	0.004	0.012	0.026	0.079
0.000	0.000	0.000	0.000	0.001	0.003	0.000	0.206
0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.588
0.000	0.000	0.000	0.000	0.000	0.008	0.190	0.000

Task 2 continued on next page. . .

Q-Learning State-Value-Function with more iterations:



0.003	0.004	0.005	0.007	0.010	0.013	0.017	0.017
0.002	0.003	0.004	0.007	0.010	0.014	0.021	0.024
0.002	0.002	0.002	0.000	0.008	0.013	0.028	0.036
0.001	0.001	0.001	0.001	0.003	0.000	0.033	0.054
0.000	0.000	0.000	0.000	0.006	0.010	0.038	0.082
0.000	0.000	0.000	0.000	0.001	0.004	0.000	0.134
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.419
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000