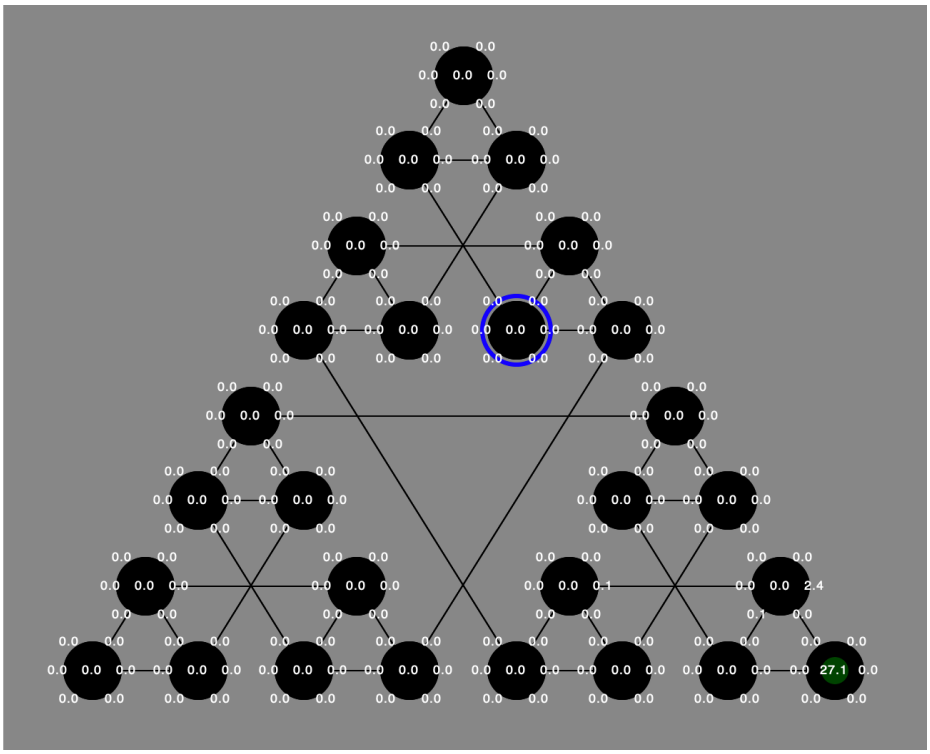
Part B Report

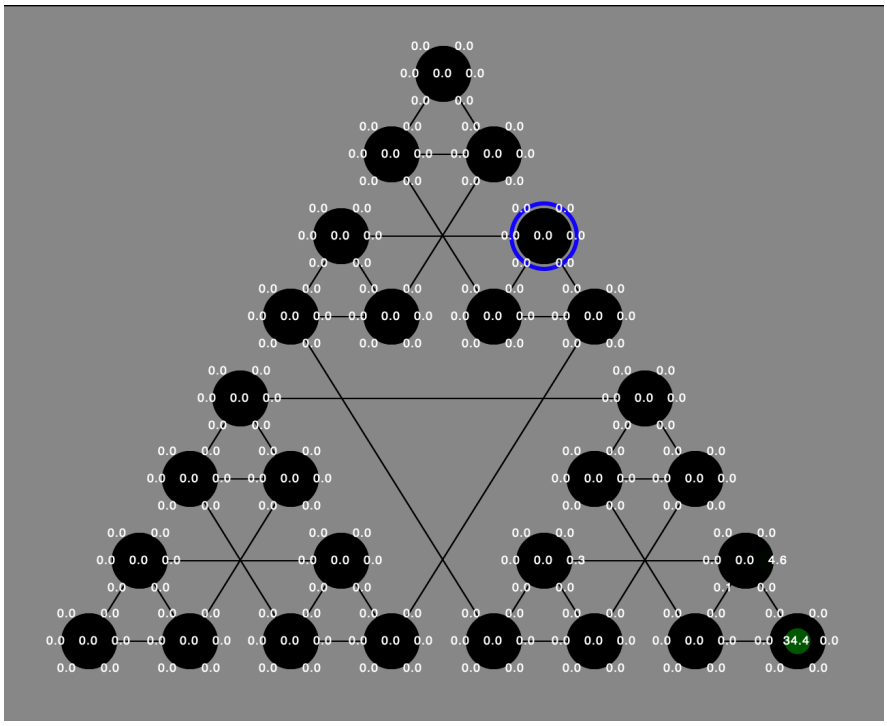
I implemented the handle_transition and choose_next_action methods.

First, I set fixed $\varepsilon = 0.1$ and $\alpha = 0.1$

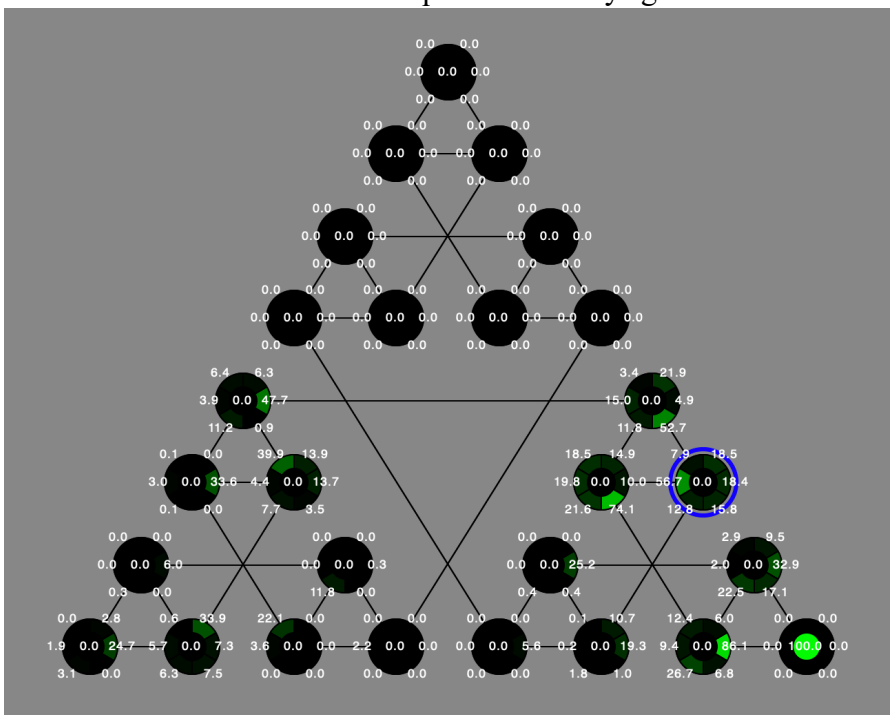
This is the after 997 runs



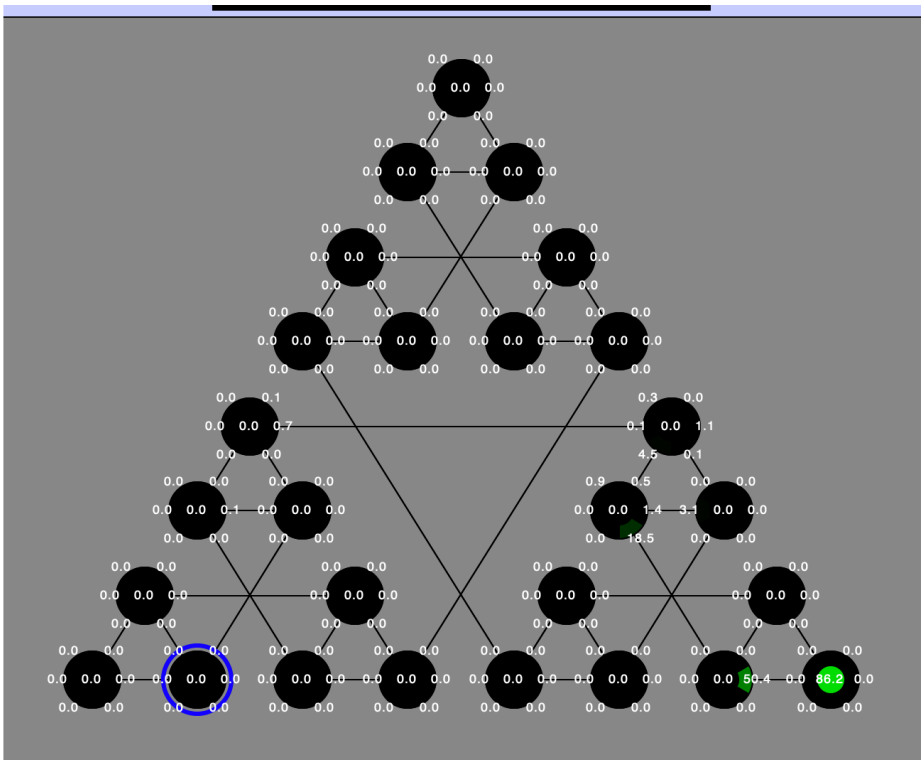
This is after 1996 runs.



After 7828 runs, I can already frequently reach the final state. But the upper triangular area is unreachable because the q value is always greater for the lower part.



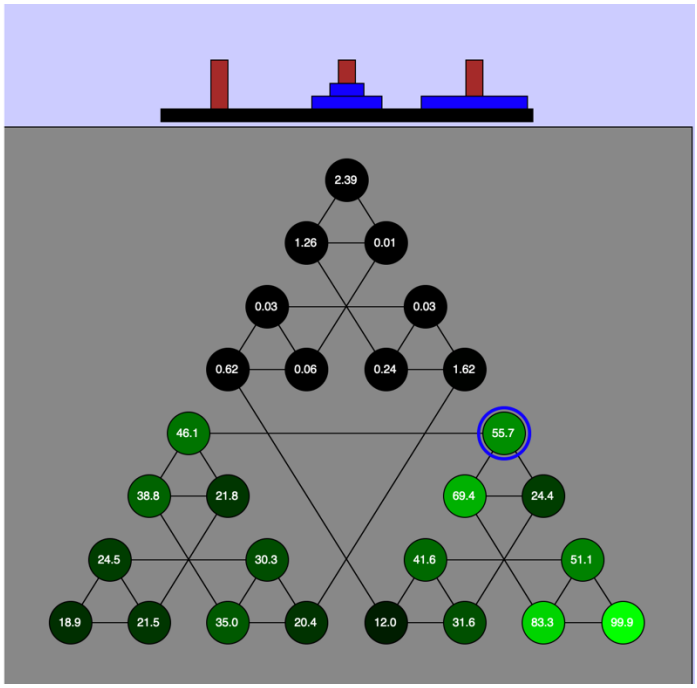
Next, I change the ε and α to custom value. According to my implementation, they will decay only when reaching the goal state. This makes sure that the learning rate and epsilon won't decrease so fast to a meaningless level before figuring out the goal state.



The effect of custom epsilon and alpha is easy to see. Only after 993 runs, the value of the goal state has already became 86.2. This is because each time after reaching the goal state, the decrease of epsilon will lead to fewer random choices and the decrease of alpha will lead to the convergence of the value. So, the program tends to keep going through the path that we already found.

Finally, I test the exploration function I wrote. For the exploration function, when updating the Q values, I'll add an extra value to it. The less it has been visited, the bigger this extra value will be. So, theoretically, the program will tends to visit states that haven't been visited before.

This is 7817 runs when doesn't use exploration functions



This is 7825 runs when using the exploration function. We can see when using exploration function, the program tends to explore more unexplored states in the state space. The speed of finding goal state do decrease, but more states are explored during the running process.