CS4408 Learning Journal 7

**Learning Journal on Markov Decision Process (MDP) Assignment**

Over the past week, I worked on an assignment focused on Markov Decision Processes (MDPs), specifically applying the concept to a grid-world scenario. In this assignment, I created a pseudo-code solution in Python-like syntax to help an agent navigate from a start position to a goal position (a diamond) while avoiding blocked cells and a fire cell.

First, I read the assignment instructions thoroughly and broke down the requirements. I identified the main tasks: designing a suitable data structure for the grid, defining the states and rewards, implementing a decision-making process based on MDP (value iteration), handling blocked and fire states, and finally testing whether the goal was reached. I then translated these tasks into a structured pseudo-code. Instead of just jumping into coding, I started with a plan: I mapped out each function I needed (e.g., is\_valid\_state, get\_next\_state\_and\_reward, value\_iteration, etc.) and determined how they would interact. This approach helped me keep the solution organized.

When I began writing the pseudo-code, I was initially concerned about making it both clear and comprehensive. My reaction to the complexity of MDPs was a mix of excitement and mild intimidation. MDPs involve iterative updates, rewards, and policy extraction, so ensuring that the value iteration logic was correct took some thought. However, once I established the basic Bellman equation—V(s)←max⁡a[R(s,a)+γ⋅V(s′)]V(s) \leftarrow \max\_{a} \big[R(s,a) + \gamma \cdot V(s')\big]V(s)←maxa​[R(s,a)+γ⋅V(s′)]—the rest became more systematic. I found it helpful to comment on each step, clarifying how each function contributed to the overall MDP solution.

Throughout the process, I participated in a discussion forum where other classmates shared their challenges. One peer pointed out that handling blocked states needed careful consideration, so I rechecked my is\_valid\_state function to ensure it properly prevented moves into blocked cells. This feedback was crucial, as it helped me refine the transitions in my code. Another peer emphasized the importance of making terminal states (fire and diamond) truly terminal by preventing further updates after they are reached, which I incorporated by adding checks in the main loop.

My feelings during this assignment oscillated between satisfaction and frustration. The satisfaction came from watching the policy iteration or value iteration converge to a sensible path. The frustration emerged when small bugs, such as indexing errors or forgetting to handle the boundary conditions, caused unexpected results. Each time I encountered an error, I stepped back, looked at the pseudo-code, and traced through a test scenario in my head (or on paper). This systematic debugging process improved my confidence in the final solution.

I learned that MDPs are powerful frameworks for decision-making under uncertainty and that even a relatively small grid-world can highlight the importance of carefully defined rewards and transition functions. From a conceptual standpoint, I deepened my understanding of the Bellman equations and the significance of discount factors (γ\gammaγ). I also gained practical insight into how blocked and fire states can be seamlessly integrated into an MDP by treating them as special cases in the transition function.

The biggest challenge was ensuring correctness in the iterative updates. Initially, I found it tempting to stop the value iteration too soon or to overlook minor differences between old and new values. However, by implementing a convergence threshold (THETA), I allowed the algorithm to iterate until changes were negligible.

In conclusion, this assignment taught me not only the technical aspects of implementing an MDP but also reinforced the value of planning, commenting, and peer feedback. It was a rewarding experience that enhanced both my theoretical understanding and my practical coding skills.

**References**  
Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press.