Mini-1

1. Figure out a unifying and practical method for recording your ideas, confusions, questions, and plans.

2. Create abstract classes and/or use inheritance for a finite Markov (plain/reward) process. Consider your data structure choices.

3. Have a method for value convergence, like we did in class, via matrix form:

$$V = \mathcal{R} + \gamma P V,$$

   or element-wise:

$$v(s) = \mathcal{R}(s) + \gamma \sum_{s' \in S} P_{ss'} \cdot v(s'),$$

   where s is a state and s' is a possible next-step state. You can use eigenvalues but be ready to deal with degenerate cases (for example, $\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$).

   When coming up with a method (or several methods) for value convergence, consider how your method scales with the number of states.

4. Create an initialization for your classes corresponding to basic grid worlds. Given 2D-masks such as

$$\begin{bmatrix} 0 & -\infty & -1 & -1 & -1 \\ -1 & -\infty & -1 & -\infty & -1 \\ -1 & -\infty & -1 & -\infty & -1 \\ -1 & -1 & -1 & -\infty & -1 \\ -3 & -3 & -3 & -3 & -3 \end{bmatrix}$$

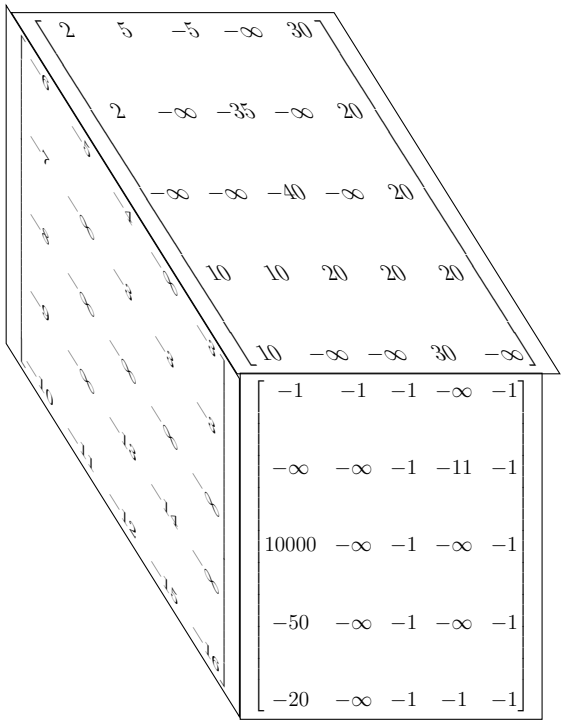
   with your coding choice in ∞, translate these masks into the corresponding data structures for your S, R and P. In particular, consider what values make sense for P and what edge cases there are to consider in general. Congrats, you now have a basic grid world to tinker around with. Experiment with long term values for states of various initial 2D-masks.

5. Is γ really necessary here? Do we need the same γ for every time step, t? Why are we multiplying by γ and not γ^2? Think it through. Experiment with different values of γ. Does it make an impact on what values converge to per state?

6. So far, there is no control over transitions and no hope that will change, regardless of how the rewards are initially given. How would you go about adding actions into the mix? Think about it.

7. (optional challenge - team + LLM friendly) Try to determine the right implementation for a complex like what is shown below without resorting to 3D initializations with

large memory footprints. See if you can further generalize your design via a cartesian product of matrix tiles along a binary tree of height 3. There is currently a real gap in the number of available gridworld options that corresponding to "Lego"-like tilings, such as what is found with trees of grids and simplicial complexes more generally. Such innovations could really boost the availability of options corresponding to reinforcement learning coupled with procedurally generated environments. Moreover, pivotal-irreversible decision making is tied to the structure of trees.



8. Give some thoughts about the whole Markov process model. Do you like it? Do you think there is a better or different way to model random decision making? Consider what you think are the advantages and limitations to markov processes. Reinforcement learning tends to focus on markov processes, but that does not mean there cannot be more innovative ideas.