HW 1: Value Iteration

In this assignment, you will implement the value iteration algorithm for an agent in a grid world environment to compute the state values and the agent policy.
Specifically, you will code the steps for (i) iterating over all the states in the grid world that the agent can be in, (ii) computing the value of each state, (iii) saving the action corresponding to the maximum value for that state, and (iv) repeating the above steps until the values converge or maximum iterations are reached.

Algorithm 1 Value Iteration Algorithm

$V_0(s) = 0$
repeat
  for $s \in S$ **do**
    $V_{t+1}(s) = \max_a [R(s) + \sum_{s'} P(s'|s,a)\gamma V_t(s')]$
  end for
until $\max |V_{t+1} - V_t| < \varepsilon$

Instructions:

- Clone the starter code from your repository to your local machine and install the requirements mentioned in the README.
- Go through the **README.md** to understand the states, actions, transitions, and rewards of the grid world environment.
- Write your value iteration code in the **vi.py** file. You have been provided methods to get the reward of a state, the actions that can be executed from that state and the possible transitions after executing an action. Refer the README for instructions on how to use the methods.
- Consider $max\_iterations = 500$, discount factor $\gamma = 0.9$ and the convergence threshold $\varepsilon = 0.001$.
- Run **main.py** to visualize the agent moving in the grid world. The main file imports your value iteration code and executes the computed optimal action from its current state until a terminal state is reached. Your agent's path will be saved to **decisions.csv**.

- Commit your edited **vi.py** and the **decisions.csv** file to your repository before the deadline.