RL Lab 2

Harsh Raj (180010017)
Akhilesh Bharadwaj (180010009)

Note: Code for all questions (including simulations etc) is attached in the submitted zip. Artifacts including the images generated are also attached in the zip. Please make sure to install dependencies mentioned in the requirements.txt before running any submitted code.

```
# Install dependencies
pip3 install -r requirements.txt

# run codes corresponding to each question
python3 runner.py
```

For the current lab, we consider the following Grid world environment as MDP:

```
   __________________
   |0  |0  |0  |1    |
   |0  |0  |0  |-100 |
   |0  |XXX|0  |0    |
   |0  |0  |0  |0    |
   -------------------
```

The cells represent reward associated with the states, `` `xxx` `` represents the state that is blocked. The states with non zero reward are absorbing states. An agent reaching an absorbing state can not move out of the state and the episode ends once an agent reaches the absorbing state.

If an agent selects an action a, there is 80% probability of the agent moving in that direction, and 10% each probability of moving in the direction orthogonal to the selected action a.

Even though this is a finite horizon setting, We formulate this as a discounted reward MDP, encouraging the agent to choose the actions that make them reach the optimal state in the least possible number of steps.

For policy and value iteration, we initialize the value functions with zeros. We observe that the policy iteration is faster to converge compared to the value iteration.

Since for an MDP, the optimal value function is unique, to confirm if the policy converged is an optimal policy, we can compare the corresponding value function with the optimal value function obtained from the value iteration.