CS747 Assignment 2

Agulla Surya Bharath **Roll.No : 17D070055**

Task 1:

Value Iteration :

I have initialized the value function vector with all ones and then ran the value iteration algorithm as mentioned in class note till the " *maximum value in the* $|V_{t+1} - V_t|$ *is less than 0.00000001* ".

Howard Policy Iteration :

While updating the policy , for each state I took the list of all improbable actions and then switched that state to choose the action which has the maximum value of the difference $\{Q^{\Pi}(s, a) - V^{\Pi}(s)\}$. I did this for all states till I reached the optimal policy without improvable actions..

In the remaining parts I just followed the way mentioned in class notes.

Linear Programming:

Here I just wrote the constraints as mentioned in the class note and then took help from the video provided on class page to use pulp.No significant extra assumptions made..

Note : After calculating optimal value function , inorder to get the optimal policy actions I have used the following relation mentioned in class note :

$$Q^{\star}(s, a) = \sum_{s' \in S} T(s, a, s')\{R(s, a, s') + \gamma V^{\star}(s')\}.$$

$$\pi^{\star}(s) = \operatorname*{argmax}_{a \in A} Q^{\star}(s, a).$$

Reference : Slide 12 of Week 5 class Slides.

Task 2 :

I have used Value iteration (vi) for optimal policy generation in maze task.

MDP Generation (encoder.py) :

I took the total number of states in "n x n" grid to be = n^2

Then added labels of 0,1,2,3 to each state as given in the problem statement..

For a move along the empty tile, I have given reward of -1/(total number of tiles)

For a move from any tile to start tile I have given a reward of -1

For a move from any tile to end tile I have given a reward of 1* (total number of tiles)

For all above mentioned transitions the transition probability is 1.

For all other remaining kind of moves I gave the transition probability = 0

So that as mentioned in the problem statement , there is no problem with invalid moves on to the wall units..

So , in the optimal policy any state will try to avoid the action that takes it to the start tile or on to the wall.

Also, while we search for an optimal policy
The large reward on end tile will attract all the states to take the actions that makes the transition new states forming a path to the end state because in finding the optimal policy , first we maximize the value function vector of each state which itself is the expected value of rewards starting from corresponding states under given policy ..

The reward of -1/(total number of tiles) for movement along empty tiles will help in finding the shortest path (i.e less number of steps maximizes the expected sum of rewards in the value function of states)

I took the discount factor (gamma) = 0.9999

planner.py (using vi) : gives us the optimal policy .

decoder.py : calculates the next state locations based on actions in optimal policy from the start state to the end state..