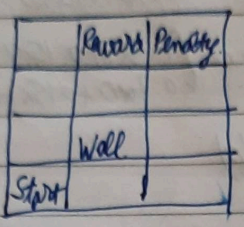
Name :- Bronav Gupta
Roll No. :- 2021101095

Machine, Data And Learning
Assignment 3

Ans Value Iteration Algorithm by Hand is as follows:-

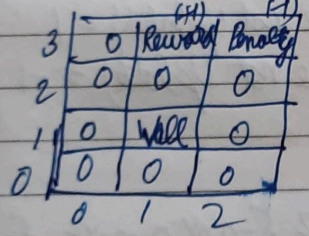
Given :- Grid (Shown Below), Step Cost = -0.04, Discount Factor(γ) = 0.95
P(going in direction of action) = 0.7, P(going in a direction ⊥ of action) = 0.15
Reward for reaching reward state = 1, Penalty for reaching penalty state = -1.



Initialize Utility Vectors dp and dp_prev (Utility of Cell in Previous Iteration as dp = [{0,0,0}, {0,0,0}, {0,0,0}, {0,1,-1}]
and dp_prev = [{0,0,0}, {0,0,0}, {0,0,0}, {0,0,0}]

Iteration 1 :-

Grid State is :-



By Bellman Update Formula,

$dp[i][j] = \text{Step Cost} + \text{Discount Factor} \times \max($
$dp[i+1][j] \times 0.7 + dp[i][j+1] \times 0.15 + dp[i][j-1] \times 0.15,$
$dp[i-1][j] \times 0.7 + dp[i][j+1] \times 0.15 + dp[i][j-1] \times 0.15,$
$dp[i][j+1] \times 0.7 + dp[i+1][j] \times 0.15 + dp[i-1][j] \times 0.15,$
$dp[i][j-1] \times 0.7 + dp[i+1][j] \times 0.15 + dp[i-1][j] \times 0.15)$

If a state is invalid, prev Utility Value of dp_prev[i][j] is considered in place of dp_prev of that state

$dp[0,0] = -0.04 + 0.95 \times \max(0.7\times0 + 0.15\times0 + 0.15\times0, 0.7\times0 + 0.15\times0 + 0.15\times0, 0.7\times0 + 0.15\times0 + 0.15\times0, 0.7\times0 + 0.15\times0 + 0.15\times0)$
$= -0.04$ (Up)

$dp[0,1] = -0.04 + 0.95 \times \max(0.7\times0 + 0.15\times0 + 0.15\times0, 0.7\times0 + 0.15\times0 + 0.15\times0, 0.7\times0 + 0.15\times0 + 0.15\times0, 0.7\times0 + 0.15\times0 + 0.15\times0)$
$= -0.04$ (Up)

$dp[0,2] = -0.04 + 0.95 \times \max(0.7\times0 + 0.15\times0 + 0.15\times0, 0.7\times0 + 0.15\times0 + 0.15\times0, 0.7\times0 + 0.15\times0 + 0.15\times0, 0.7\times0 + 0.15\times0 + 0.15\times0)$
$= -0.04$ (Up)

dp[1,0] = -0.04 + 0.95 x max(0.7x0 + 0.15x0 + 0.15x0, 0.7x0 + 0.15x0 + 0.15x0,
0.7x0 + 0.15x0 + 0.15x0, 0.7x0 + 0.15x0 + 0.15x0)
= -0.04 (Up)

dp[1,1] = Wall (Utility Value not Updated) = 0

dp[1,2] = -0.04 + 0.95 x max(0.7x0 + 0.15x0 + 0.15x0, 0.7x0 + 0.15x0 + 0.15x0
0.7x0 + 0.15x0 + 0.15x0, 0.7x0 + 0.15x0 + 0.15x0)
= -0.04 (Up)

dp[2,0] = -0.04 + 0.95 x max(0.7x0 + 0.15x0 + 0.15x0, 0.7x0 + 0.15x0 + 0.15x0
0.7x0 + 0.15x0 + 0.15x0, 0.7x0 + 0.15x0 + 0.15x0)
= -0.04 (Up)

dp[3,1] = -0.04 + 0.95 x max(0.7x1 + 0.15x0 + 0.15x0, 0.7x0 + 0.15x0 + 0.15x0,
0.7x0 + 0.15x1 + 0.15x0, 0.7x0 + 0.15x1 + 0.15x0)
= -0.04 + 0.95 x 0.7
= 0.625 (Up)

dp[3,2] = -0.04 + 0.95 x max(0.7x(-1) + 0.15x0 + 0.15x0, 0.7x0 + 0.15x0 + 0.15x0,
0.7x0 + 0.15x(-1) + 0.15x0, 0.7x0 + 0.15x(-1) + 0.15x0)
= -0.04 (Down)

dp[3,0] = -0.04 + 0.95 x max(0.7x0 + 0.15x0 + 0.15x1, 0.7x0 + 0.15x0 + 0.15x1,
0.7x1 + 0.15x0 + 0.15x1, 0.7x0 + 0.15x0 + 0.15x0)
= -0.04 + 0.95 x 0.7
= 0.625 (Right)

dp[3,1] = Reward (Utility Value not Updated) = 1

dp[3,2] = Penalty (Utility Value not Updated) = -1

Set dp_prev = dp

⇒ dp_prev = [{ -0.04, -0.04, -0.04 }, { -0.04, 0, -0.04 }, { -0.04, 0.625, -0.04 },
{ 0.625, 1, -1 }]

Iteration 2:-

Grid State is :-

0.625	Reward (+1)	Penalty (-1)
-0.04	0.625	-0.04
-0.04	Wall	-0.04
-0.04	-0.04	-0.04

Using the Bellman Update Formula, we get

$dp[0][0] = -0.04 + 0.95 \times \max($
$0.7(-0.04) + 0.15(-0.04) + 0.15(-0.04),$
$0.7(-0.04) + 0.15(-0.04) + 0.15(-0.04),$
$0.7(-0.04) + 0.15(-0.04) + 0.15(-0.04),$
$0.7(-0.04) + 0.15(-0.04) + 0.15(-0.04))$

$= -0.04 + 0.95 \times (-0.04)$

$= -0.04 - 0.038 = -0.078$ (Up)

$dp[0][1] = -0.04 + 0.95 \times \max($
$0.7(-0.04) + 0.15(-0.04) + 0.15(-0.04),$
$0.7(-0.04) + 0.15(-0.04) + 0.15(-0.04),$
$0.7(-0.04) + 0.15(-0.04) + 0.15(-0.04),$
$0.7(-0.04) + 0.15(-0.04) + 0.15(-0.04))$

$= -0.04 + 0.95(-0.04)$

$= -0.078$ (Up)

$dp[0][2] = -0.04 + 0.95 \times \max($
$0.7(-0.04) + 0.15(-0.04) + 0.15(-0.04),$
$0.7(-0.04) + 0.15(-0.04) + 0.15(-0.04),$
$0.7(-0.04) + 0.15(-0.04) + 0.15(-0.04),$
$0.7(-0.04) + 0.15(-0.04) + 0.15(-0.04))$

$= -0.04 + 0.95(-0.04)$

$= -0.078$ (Up)

$dp[1][0] = -0.04 + 0.95 \times \max($
$0.7(-0.04) + 0.15(-0.04) + 0.15(-0.04),$
$0.7(-0.04) + 0.15(-0.04) + 0.15(-0.04),$
$0.7(-0.04) + 0.15(-0.04) + 0.15(-0.04),$
$0.7(-0.04) + 0.15(-0.04) + 0.15(-0.04))$

$= -0.04 + 0.95(-0.04)$

$= -0.078$ (Up)

$dp[1][1] =$ Wall (utility value not updated) $= 0$

$dp[1][2] = -0.04 + 0.95 \times \max($
$0.7(-0.04) + 0.15(-0.04) + 0.15(-0.04),$
$0.7(-0.04) + 0.15(-0.04) + 0.15(-0.04),$
$0.7(-0.04) + 0.15(-0.04) + 0.15(-0.04),$
$0.7(-0.04) + 0.15(-0.04) + 0.15(-0.04))$

$= -0.04 + 0.95(-0.04)$

$= -0.078$ (Up)

$dp[2][0] = -0.04 + 0.95 \times \max(0.7\times0.625 + 0.15\times0.625 + 0.15\times(-0.04),$
$0.7\times(-0.04) + 0.15\times0.625 + 0.15\times(-0.04),$
$0.7\times(-0.04) + 0.15\times0.625 + 0.15\times(-0.04),$
$0.7\times(0.625) + 0.15\times0.625 + 0.15\times(-0.04))$

$= -0.04 + 0.95 \times 0.52525$

$= 0.458975 \cong 0.459$ (Right)

$dp[2][1] = -0.04 + 0.95 \times \max(0.7\times1 + 0.15\times(-0.04) + 0.15\times(-0.04),$
$0.7\times0 + 0.15\times(-0.04) + 0.15\times(-0.04),$
$0.7\times(-0.04) + 0.15\times1 + 0.15\times0,$
$0.7\times(-0.04) + 0.15\times1 + 0.15\times0)$

$= -0.04 + 0.95 \times 0.688$

$= 0.6136$ (Up)

$dp[2][2] = -0.04 + 0.95 \times \max(0.7\times(-1) + 0.15\times0.625 + 0.15\times(-0.04),$
$0.7\times(-0.04) + 0.15\times0.625 + 0.15\times(-0.04),$
$0.7\times0.625 + 0.15\times(-1) + 0.15\times(-0.04),$
$0.7\times(-0.04) + 0.15\times(-1) + 0.15\times(-0.04))$

$= -0.04 + 0.95 \times 0.2815$

$= 0.227425 \cong 0.227$ (left)

$dp[3][0] = -0.04 + 0.95 \times \max(0.7\times0.625 + 0.15\times1 + 0.15\times0.625,$
$0.7\times(-0.04) + 0.15\times1 + 0.15\times0.625,$
$0.7\times(0.625) + 0.15\times0.625 + 0.15\times(-0.04),$
$0.7\times1 + 0.15\times0.625 + 0.15\times(-0.04))$

$= -0.04 + 0.95 \times 0.738625$

$= 0.708625 \cong 0.708$ (Right)

$dp[3][1] =$ Reward (Utility value not Updated) $= 1$

$dp[3][2] =$ Penalty (Utility value not Updated) $= -1$

On comparing the values obtained in first 2 iterations by running the program, we observe that values obtained are more or less the same and minor fluctuations are caused by Rounding Off Error in Multiplication in Python.

The Grid State is

0.208	Reward	Penalty
0459	0.6136	0.227
-0.078	Wall	-0.078
-0.078	-0.078	-0.078