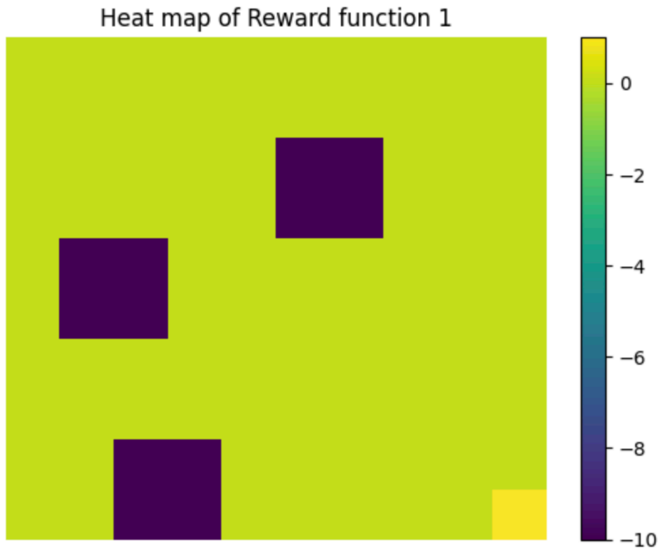
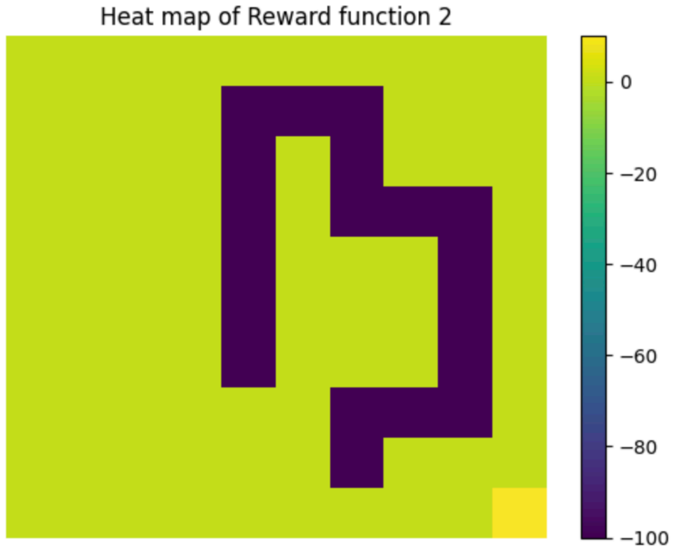
Project 3_Reinforcement learning and Inverse Reinforcement learning
Haoting Ni (905545789), Yikai Wang (905522085), Yuanxuan Fang (005949389)

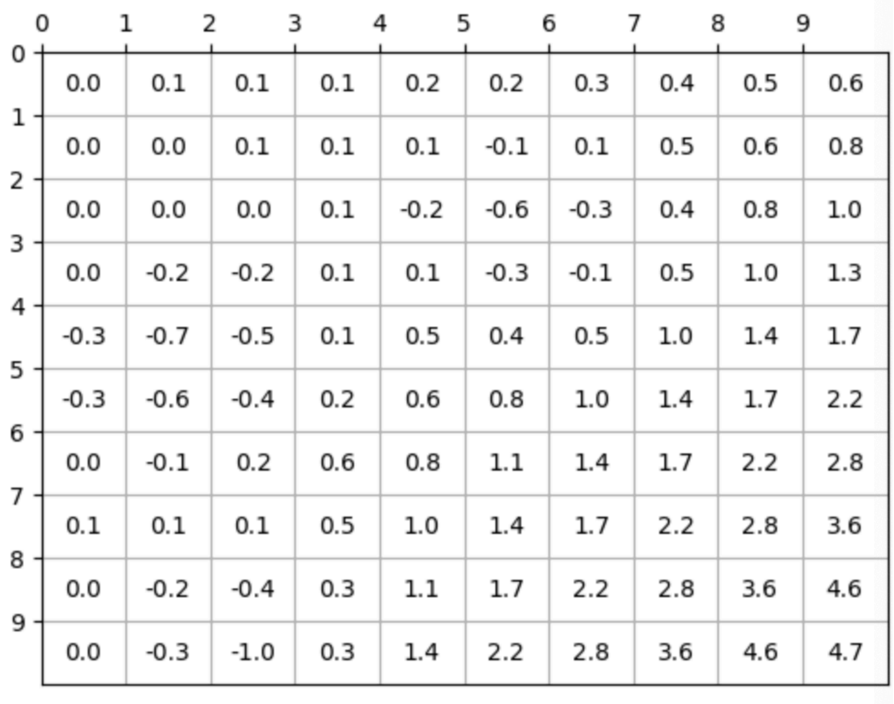
Question 1:





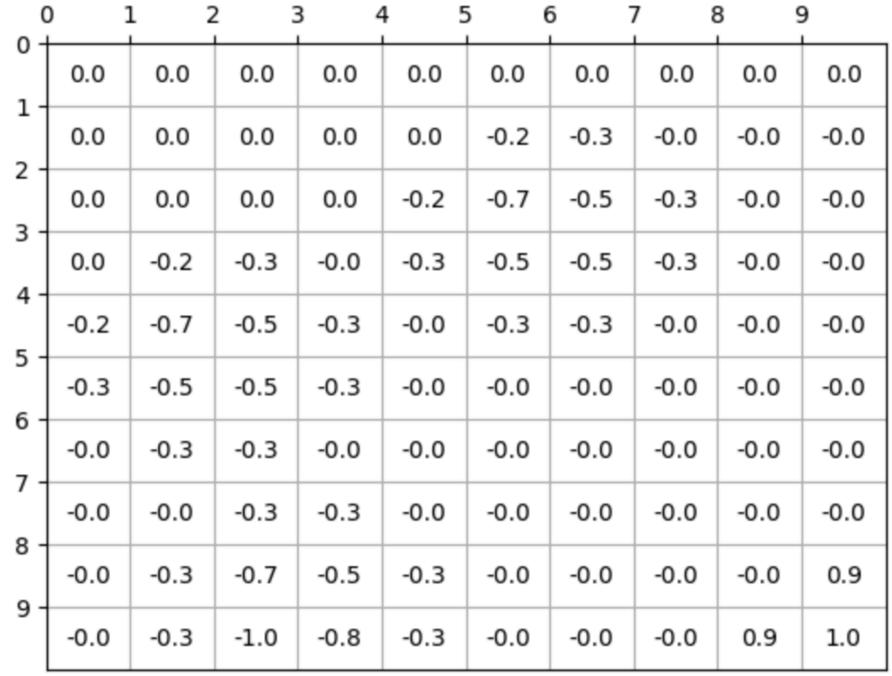
Question 2:

Optimal value of each state for reward 1



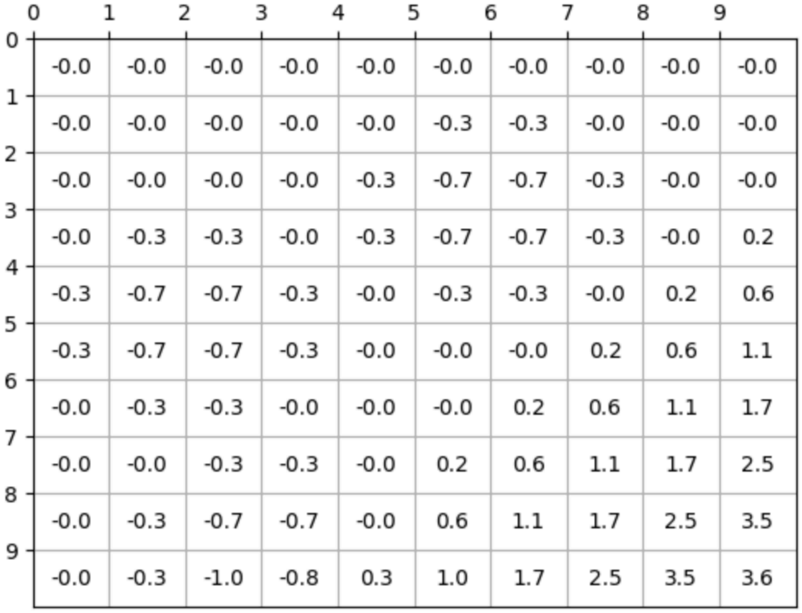
	0	1	2	3	4	5	6	7	8	9
0	0.0	0.1	0.1	0.1	0.2	0.2	0.3	0.4	0.5	0.6
1	0.0	0.0	0.1	0.1	0.1	-0.1	0.1	0.5	0.6	0.8
2	0.0	0.0	0.0	0.1	-0.2	-0.6	-0.3	0.4	0.8	1.0
3	0.0	-0.2	-0.2	0.1	0.1	-0.3	-0.1	0.5	1.0	1.3
4	-0.3	-0.7	-0.5	0.1	0.5	0.4	0.5	1.0	1.4	1.7
5	-0.3	-0.6	-0.4	0.2	0.6	0.8	1.0	1.4	1.7	2.2
6	0.0	-0.1	0.2	0.6	0.8	1.1	1.4	1.7	2.2	2.8
7	0.1	0.1	0.1	0.5	1.0	1.4	1.7	2.2	2.8	3.6
8	0.0	-0.2	-0.4	0.3	1.1	1.7	2.2	2.8	3.6	4.6
9	0.0	-0.3	-1.0	0.3	1.4	2.2	2.8	3.6	4.6	4.7

```
Step:  1
```



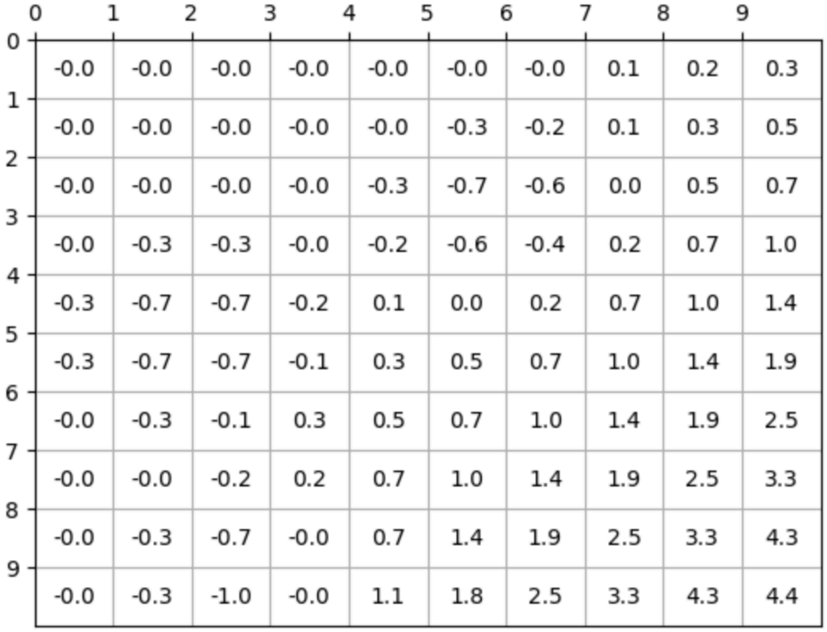
	0	1	2	3	4	5	6	7	8	9
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	-0.2	-0.3	-0.0	-0.0	-0.0
2	0.0	0.0	0.0	0.0	-0.2	-0.7	-0.5	-0.3	-0.0	-0.0
3	0.0	-0.2	-0.3	-0.0	-0.3	-0.5	-0.5	-0.3	-0.0	-0.0
4	-0.2	-0.7	-0.5	-0.3	-0.0	-0.3	-0.3	-0.0	-0.0	-0.0
5	-0.3	-0.5	-0.5	-0.3	-0.0	-0.0	-0.0	-0.0	-0.0	-0.0
6	-0.0	-0.3	-0.3	-0.0	-0.0	-0.0	-0.0	-0.0	-0.0	-0.0
7	-0.0	-0.0	-0.3	-0.3	-0.0	-0.0	-0.0	-0.0	-0.0	-0.0
8	-0.0	-0.3	-0.7	-0.5	-0.3	-0.0	-0.0	-0.0	-0.0	0.9
9	-0.0	-0.3	-1.0	-0.8	-0.3	-0.0	-0.0	-0.0	0.9	1.0

Step: 6



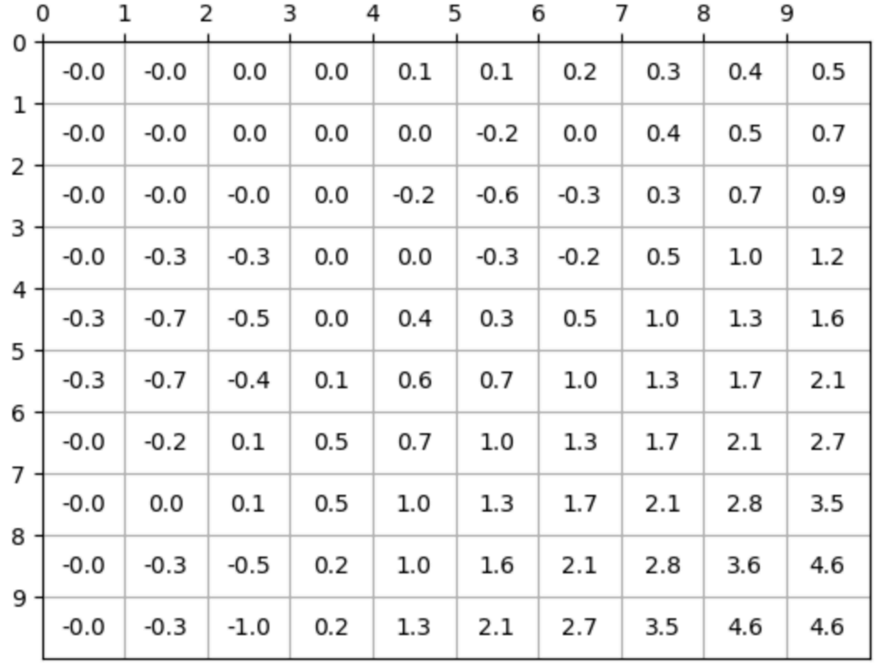
	0	1	2	3	4	5	6	7	8	9
0	-0.0	-0.0	-0.0	-0.0	-0.0	-0.0	-0.0	-0.0	-0.0	-0.0
1	-0.0	-0.0	-0.0	-0.0	-0.0	-0.3	-0.3	-0.0	-0.0	-0.0
2	-0.0	-0.0	-0.0	-0.0	-0.3	-0.7	-0.7	-0.3	-0.0	-0.0
3	-0.0	-0.3	-0.3	-0.0	-0.3	-0.7	-0.7	-0.3	-0.0	0.2
4	-0.3	-0.7	-0.7	-0.3	-0.0	-0.3	-0.3	-0.0	0.2	0.6
5	-0.3	-0.7	-0.7	-0.3	-0.0	-0.0	-0.0	0.2	0.6	1.1
6	-0.0	-0.3	-0.3	-0.0	-0.0	-0.0	0.2	0.6	1.1	1.7
7	-0.0	-0.0	-0.3	-0.3	-0.0	0.2	0.6	1.1	1.7	2.5
8	-0.0	-0.3	-0.7	-0.7	-0.0	0.6	1.1	1.7	2.5	3.5
9	-0.0	-0.3	-1.0	-0.8	0.3	1.0	1.7	2.5	3.5	3.6

Step: 11



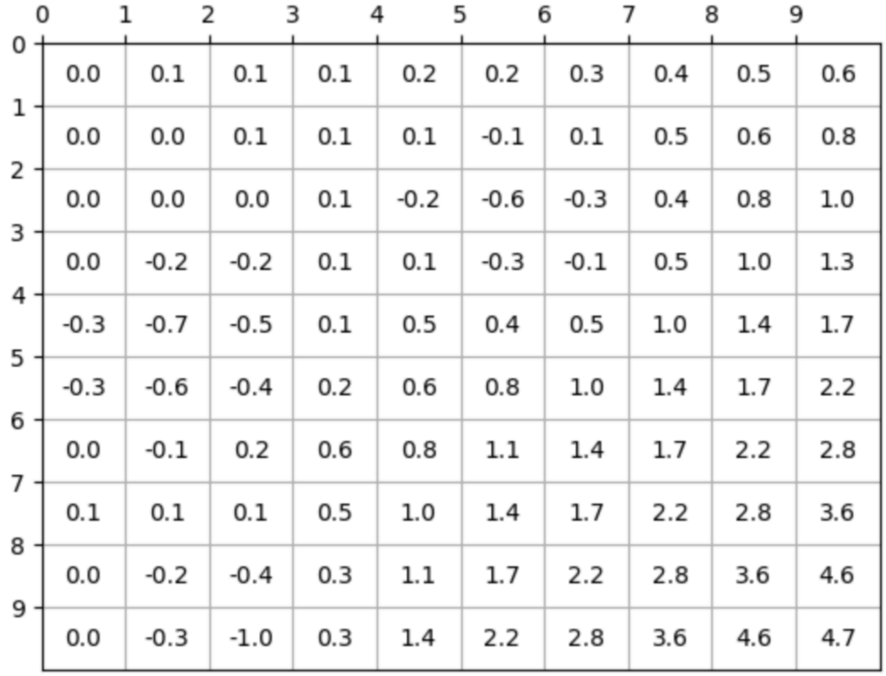
	0	1	2	3	4	5	6	7	8	9
0	-0.0	-0.0	-0.0	-0.0	-0.0	-0.0	-0.0	0.1	0.2	0.3
1	-0.0	-0.0	-0.0	-0.0	-0.0	-0.3	-0.2	0.1	0.3	0.5
2	-0.0	-0.0	-0.0	-0.0	-0.3	-0.7	-0.6	0.0	0.5	0.7
3	-0.0	-0.3	-0.3	-0.0	-0.2	-0.6	-0.4	0.2	0.7	1.0
4	-0.3	-0.7	-0.7	-0.2	0.1	0.0	0.2	0.7	1.0	1.4
5	-0.3	-0.7	-0.7	-0.1	0.3	0.5	0.7	1.0	1.4	1.9
6	-0.0	-0.3	-0.1	0.3	0.5	0.7	1.0	1.4	1.9	2.5
7	-0.0	-0.0	-0.2	0.2	0.7	1.0	1.4	1.9	2.5	3.3
8	-0.0	-0.3	-0.7	-0.0	0.7	1.4	1.9	2.5	3.3	4.3
9	-0.0	-0.3	-1.0	-0.0	1.1	1.8	2.5	3.3	4.3	4.4

```
Step:  16
```



	0	1	2	3	4	5	6	7	8	9
0	-0.0	-0.0	0.0	0.0	0.1	0.1	0.2	0.3	0.4	0.5
1	-0.0	-0.0	0.0	0.0	0.0	-0.2	0.0	0.4	0.5	0.7
2	-0.0	-0.0	-0.0	0.0	-0.2	-0.6	-0.3	0.3	0.7	0.9
3	-0.0	-0.3	-0.3	0.0	0.0	-0.3	-0.2	0.5	1.0	1.2
4	-0.3	-0.7	-0.5	0.0	0.4	0.3	0.5	1.0	1.3	1.6
5	-0.3	-0.7	-0.4	0.1	0.6	0.7	1.0	1.3	1.7	2.1
6	-0.0	-0.2	0.1	0.5	0.7	1.0	1.3	1.7	2.1	2.7
7	-0.0	0.0	0.1	0.5	1.0	1.3	1.7	2.1	2.8	3.5
8	-0.0	-0.3	-0.5	0.2	1.0	1.6	2.1	2.8	3.6	4.6
9	-0.0	-0.3	-1.0	0.2	1.3	2.1	2.7	3.5	4.6	4.6

```
Step:  21
```

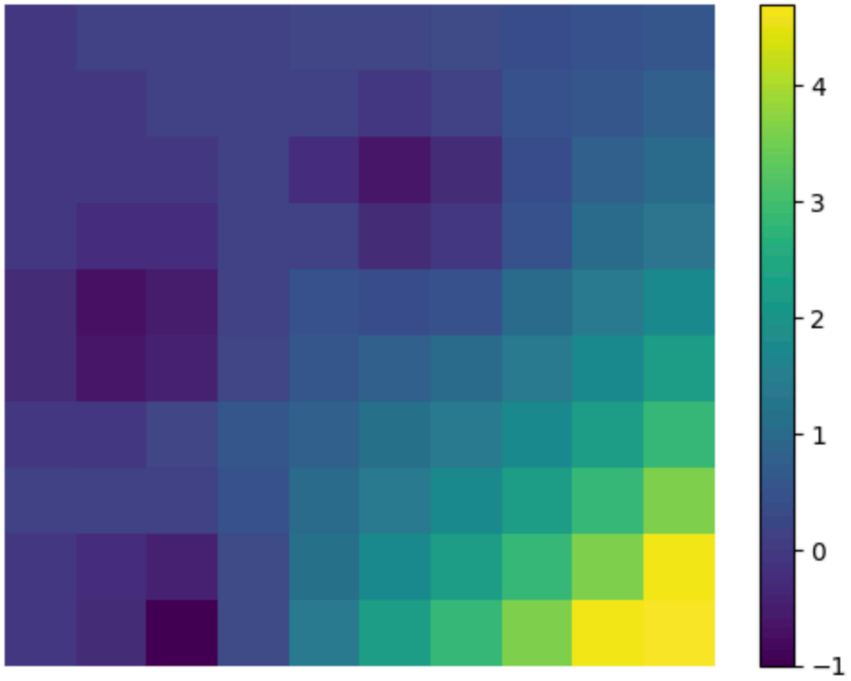


	0	1	2	3	4	5	6	7	8	9
0	0.0	0.1	0.1	0.1	0.2	0.2	0.3	0.4	0.5	0.6
1	0.0	0.0	0.1	0.1	0.1	-0.1	0.1	0.5	0.6	0.8
2	0.0	0.0	0.0	0.1	-0.2	-0.6	-0.3	0.4	0.8	1.0
3	0.0	-0.2	-0.2	0.1	0.1	-0.3	-0.1	0.5	1.0	1.3
4	-0.3	-0.7	-0.5	0.1	0.5	0.4	0.5	1.0	1.4	1.7
5	-0.3	-0.6	-0.4	0.2	0.6	0.8	1.0	1.4	1.7	2.2
6	0.0	-0.1	0.2	0.6	0.8	1.1	1.4	1.7	2.2	2.8
7	0.1	0.1	0.1	0.5	1.0	1.4	1.7	2.2	2.8	3.6
8	0.0	-0.2	-0.4	0.3	1.1	1.7	2.2	2.8	3.6	4.6
9	0.0	-0.3	-1.0	0.3	1.4	2.2	2.8	3.6	4.6	4.7

According to the implementation, line 10, V(s) which is the optiminzed value we try to get is defined by the average discounted return for taking action a in state s by following policy pi. The goal to this is that the agent tries to get the maximized expected value by taking action. As step number increases, the state value become less sparse. For example, in the early stage step 1, most value are 0, as steps go up, the value tend to be the optimal value.

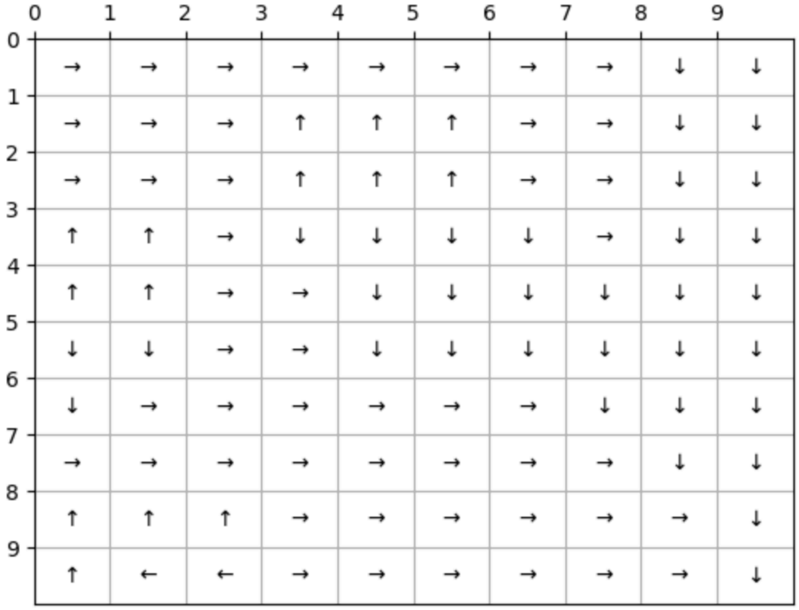
Question 3:

Heat map of optimal state values for Reward function 1



Question 4:
As the heat map of optimal state values for reward function 1 shows, if the reward increases, the optimal state values will be higher, and the color gets lighter. For example, the right down corner has the reward of 1, which means it has the highest optimal state values, and it has the lightest color in the plot. Whereas, the left up corner has the reward of 0, it has the lowest optimal state values, and it has the darkest color in the plot.

Question 5:

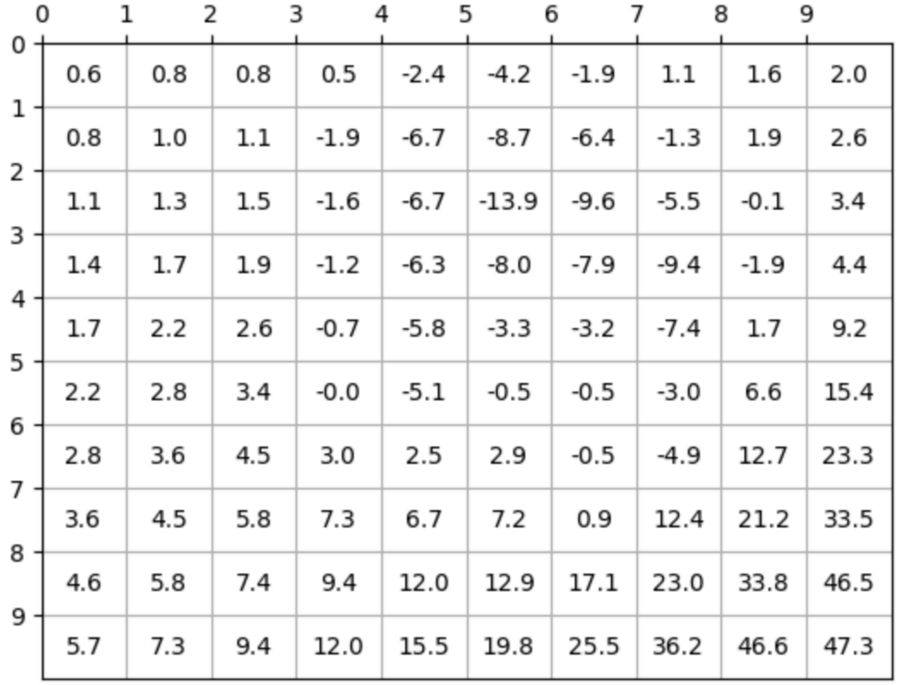


	0	1	2	3	4	5	6	7	8	9
0	→	→	→	→	→	→	→	→	↓	↓
1	→	→	→	↑	↑	↑	→	→	↓	↓
2	→	→	→	↑	↑	↑	→	→	↓	↓
3	↑	↑	→	↓	↓	↓	↓	→	↓	↓
4	↑	↑	→	→	↓	↓	↓	↓	↓	↓
5	↓	↓	→	→	↓	↓	↓	↓	↓	↓
6	↓	→	→	→	→	→	→	↓	↓	↓
7	→	→	→	→	→	→	→	→	↓	↓
8	↑	↑	↑	→	→	→	→	→	→	↓
9	↑	←	←	→	→	→	→	→	→	↓

The optimal policy of the agent matches the intuition. The agent would move with the aim of highest reward states. Thus, from the heat map, the agent should move away from the dark color area to the light color area to get the optimal value of rewards. In the end, the agent has a trend to get to the right down corner.

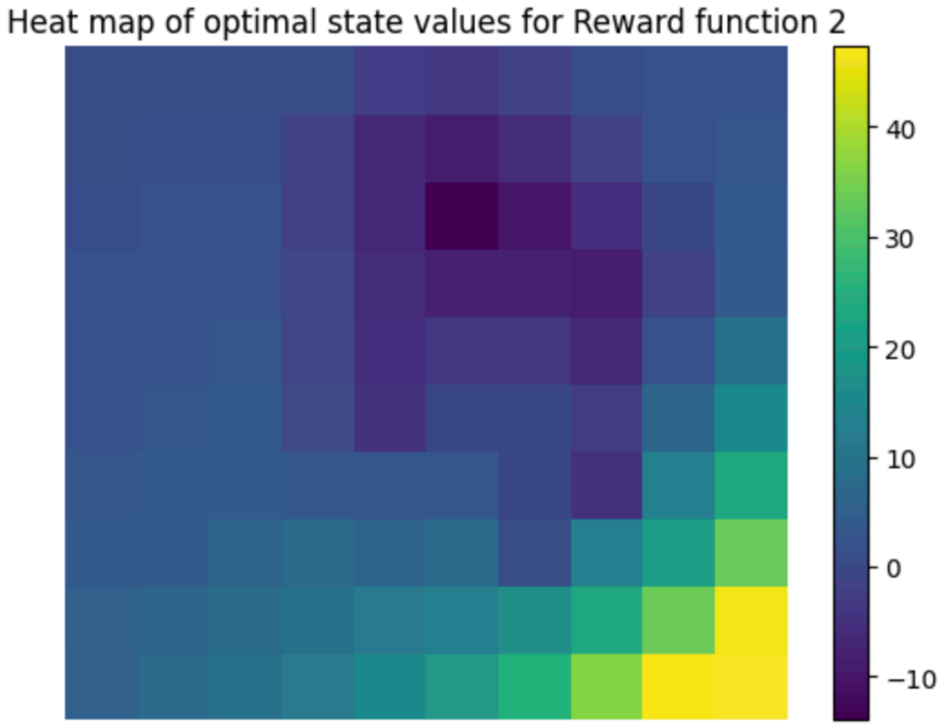
It is possible for the agent to compute the optimal action to take at each state by observing the optimal values of it’s neighboring states because besides the goal state, each state has a equal values reward function. In the iteration algorithm, we consider optimal value of neighboring states to take the move to get the optimal rewards.

Question 6:
Optimal value of each state for reward 2



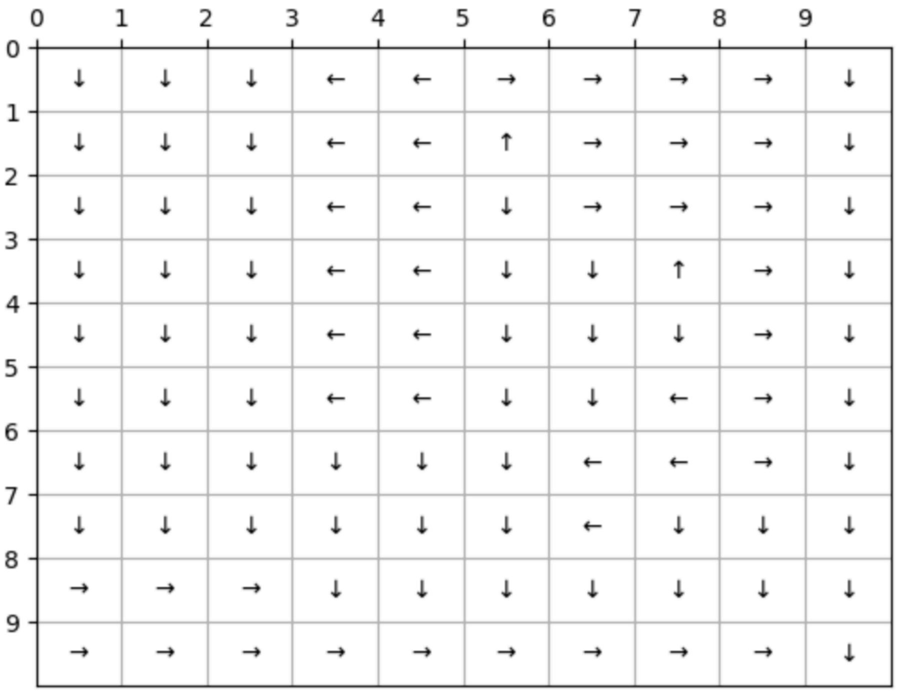
	0	1	2	3	4	5	6	7	8	9
0	0.6	0.8	0.8	0.5	-2.4	-4.2	-1.9	1.1	1.6	2.0
1	0.8	1.0	1.1	-1.9	-6.7	-8.7	-6.4	-1.3	1.9	2.6
2	1.1	1.3	1.5	-1.6	-6.7	-13.9	-9.6	-5.5	-0.1	3.4
3	1.4	1.7	1.9	-1.2	-6.3	-8.0	-7.9	-9.4	-1.9	4.4
4	1.7	2.2	2.6	-0.7	-5.8	-3.3	-3.2	-7.4	1.7	9.2
5	2.2	2.8	3.4	-0.0	-5.1	-0.5	-0.5	-3.0	6.6	15.4
6	2.8	3.6	4.5	3.0	2.5	2.9	-0.5	-4.9	12.7	23.3
7	3.6	4.5	5.8	7.3	6.7	7.2	0.9	12.4	21.2	33.5
8	4.6	5.8	7.4	9.4	12.0	12.9	17.1	23.0	33.8	46.5
9	5.7	7.3	9.4	12.0	15.5	19.8	25.5	36.2	46.6	47.3

Question 7:


Heat map of optimal state values for Reward function 2

It basically has the similar result from what we get from question 4. As the heat map of optimal state values for reward function 2 shows, if the reward increases, the optimal state values will be higher, and the color gets lighter. For example, the right down corner has the reward of 1, which means it has the highest optimal state values, and it has the lightest color in the plot. Whereas, the left up corner has the reward of 0, it has the lowest optimal state values, and it has the darkest color in the plot.

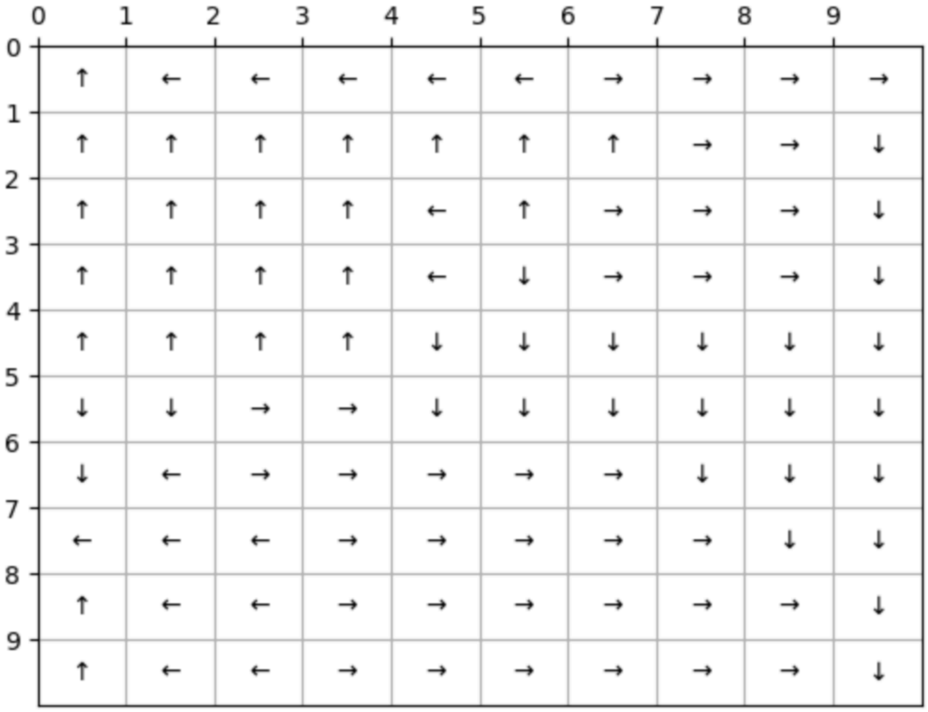
Question 8:
Optimal policy for reward function 2



	0	1	2	3	4	5	6	7	8	9
0	↓	↓	↓	←	←	→	→	→	→	↓
1	↓	↓	↓	←	←	↑	→	→	→	↓
2	↓	↓	↓	←	←	↓	→	→	→	↓
3	↓	↓	↓	←	←	↓	↓	↑	→	↓
4	↓	↓	↓	←	←	↓	↓	↓	→	↓
5	↓	↓	↓	←	←	↓	↓	←	→	↓
6	↓	↓	↓	↓	↓	↓	←	←	→	↓
7	↓	↓	↓	↓	↓	↓	←	↓	↓	↓
8	→	→	→	↓	↓	↓	↓	↓	↓	↓
9	→	→	→	→	→	→	→	→	→	↓

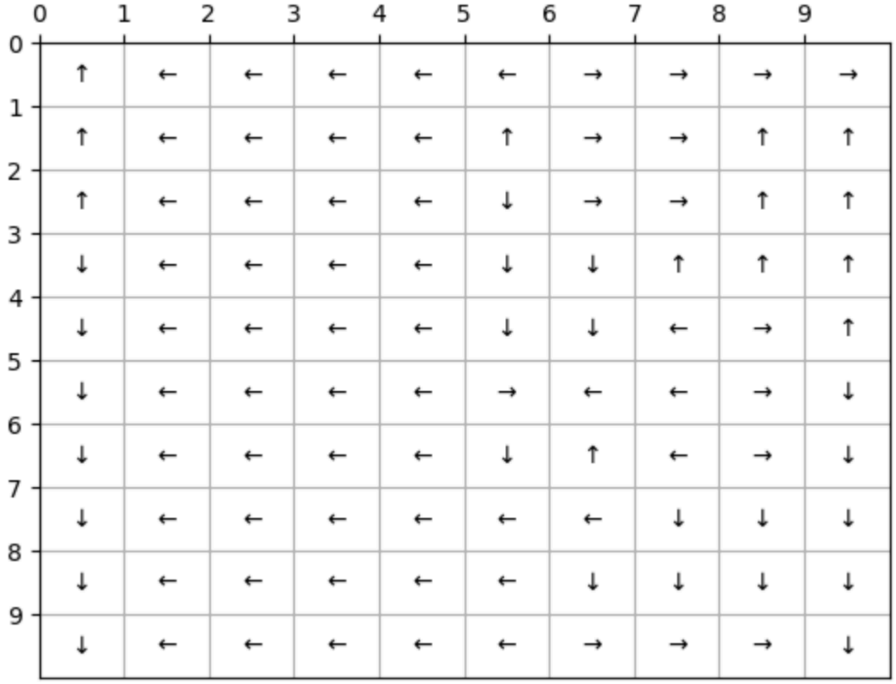
The optimal policy of the agent matches the intuition. The agent would move with the aim of highest reward states. Thus, from the heat map at question 7, the agent should move away from the dark color area to the light color area to get the optimal value of rewards. In the end, the agent has a trend to get to the right down corner.

Question 9:
Optimal policy for reward function 1, w=0.6



	0	1	2	3	4	5	6	7	8	9
0	↑	←	←	←	←	←	→	→	→	→
1	↑	↑	↑	↑	↑	↑	↑	→	→	↓
2	↑	↑	↑	↑	←	↑	→	→	→	↓
3	↑	↑	↑	↑	←	↓	→	→	→	↓
4	↑	↑	↑	↑	↓	↓	↓	↓	↓	↓
5	↓	↓	→	→	↓	↓	↓	↓	↓	↓
6	↓	←	→	→	→	→	→	↓	↓	↓
7	←	←	←	→	→	→	→	→	↓	↓
8	↑	←	←	→	→	→	→	→	→	↓
9	↑	←	←	→	→	→	→	→	→	↓

Optimal policy for reward function 2, w=0.6



	0	1	2	3	4	5	6	7	8	9
0	↑	←	←	←	←	←	→	→	→	→
1	↑	←	←	←	←	↑	→	→	↑	↑
2	↑	←	←	←	←	↓	→	→	↑	↑
3	↓	←	←	←	←	↓	↓	↑	↑	↑
4	↓	←	←	←	←	↓	↓	←	→	↑
5	↓	←	←	←	←	→	←	←	→	↓
6	↓	←	←	←	←	↓	↑	←	→	↓
7	↓	←	←	←	←	←	←	↓	↓	↓
8	↓	←	←	←	←	←	↓	↓	↓	↓
9	↓	←	←	←	←	←	→	→	→	↓

The exploration parameter, denoted as w, regulates the level of exploration performed by the agent. Increasing the value of w leads to greater exploration and more frequent random actions taken by the agent. As we can see from the plot, w increases from 0.1 to 0.6, the agent does more amount of moves. With w = 0.6, the agent take more random actions, which also means that there are more spreading out, more stakes will be taken. Thus, with w =0.1 is better for the optimal policy because it will do the action with less random actions and it will achieve the optimal value state quicker.

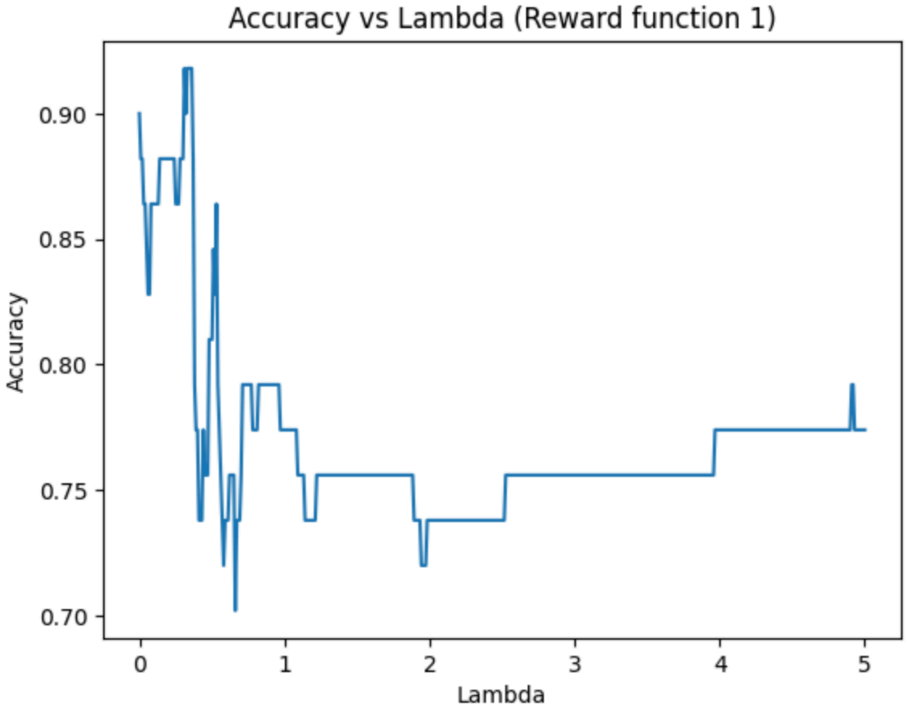
Question 10:

$$x = \begin{bmatrix} t \\ u \\ R \end{bmatrix}, c = \begin{bmatrix} 1_{|s| \times 1} \\ -\lambda \cdot 1_{|s| \times 1} \\ 0_{|s| \times 1} \end{bmatrix}$$

$$D = \begin{bmatrix} I_{|s|\times|s|} & 0 & \left(P_{(a)} - P_{(a1)}\right)\left(I - \gamma P_{(a1)}\right)^{-1} \\ 0 & 0 & \left(P_{(a)} - P_{(a1)}\right)\left(I - \gamma P_{(a1)}\right)^{-1} \\ 0 & -I_{|s|\times|s|} & I_{|s|\times|s|} \\ 0 & -I_{|s|\times|s|} & -I_{|s|\times|s|} \\ 0 & 0 & I_{|s|\times|s|} \\ 0 & 0 & -I_{|s|\times|s|} \end{bmatrix}$$

$$b = \begin{bmatrix} 0_{|s| \times 1} \\ 0_{|s| \times 1} \\ 0_{|s| \times 1} \\ 0_{|s| \times 1} \\ Rmax_{|s| \times 1} \\ Rmax_{|s| \times 1} \end{bmatrix}$$

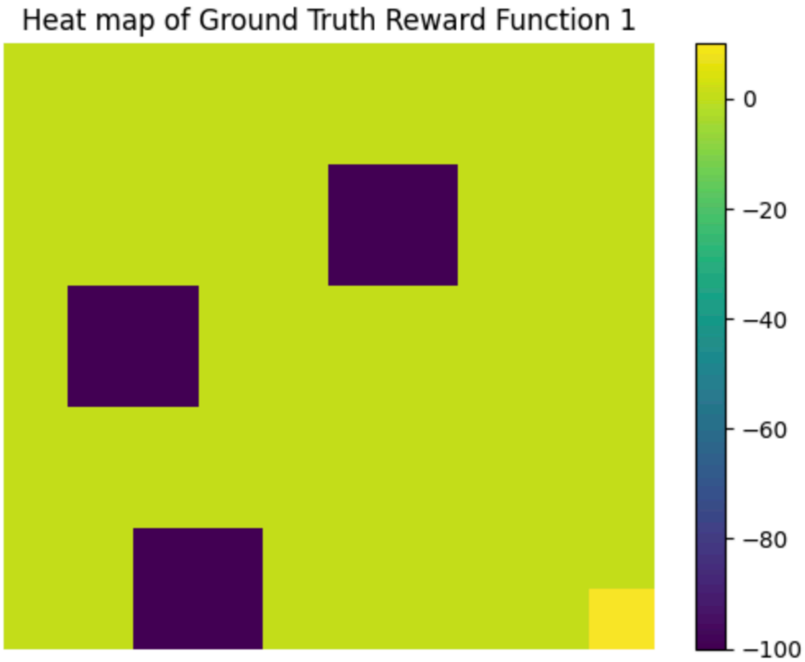
Question 11:



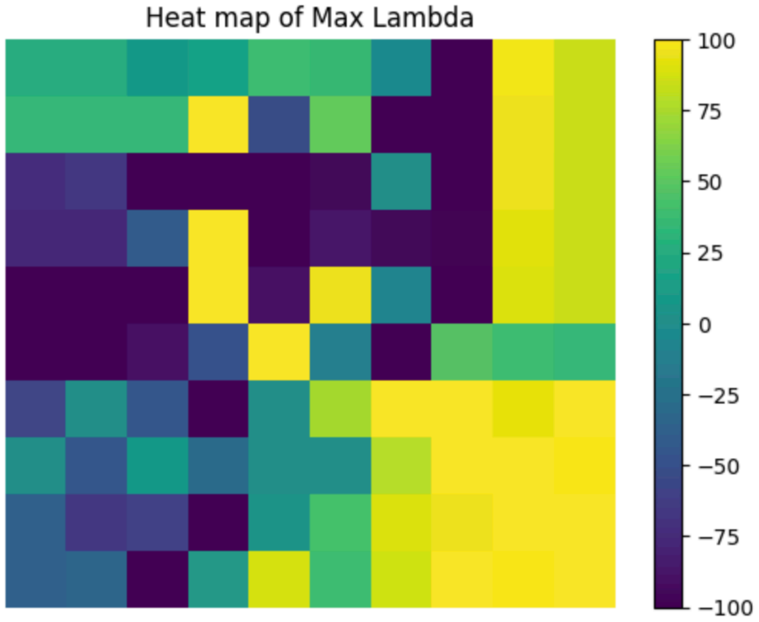
Question 12:

The best lamda $\lambda^{(1)}_{max} = 0.3106212424849699$ with accuracy of 0.918

Question 13:

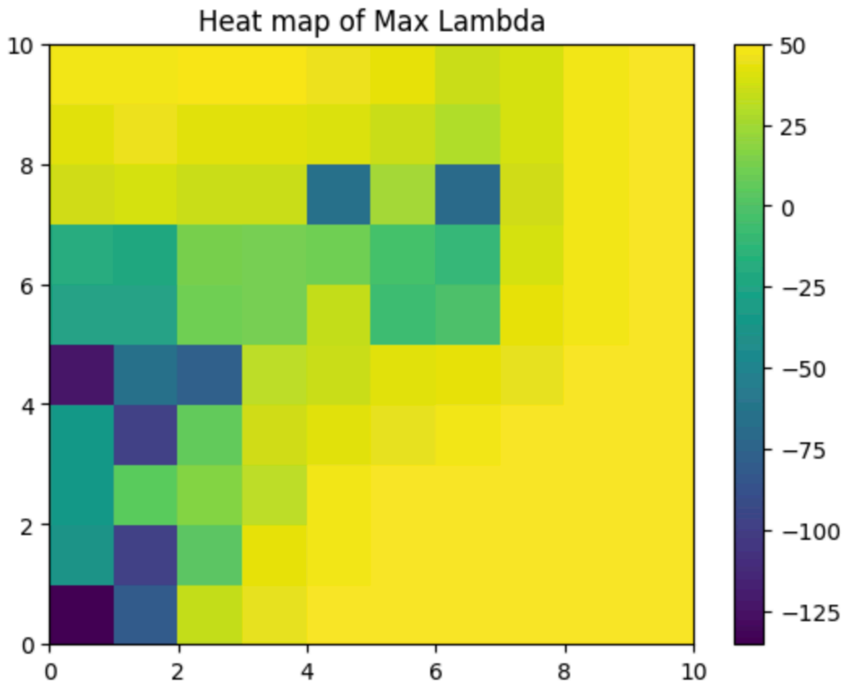


Heat map of Ground Truth Reward Function 1



Heat map of Max Lambda

Question 14:



Question 15:
Similarities:

Both has the highest value state at right down coner, state 99. Lowest value state at top left corner.

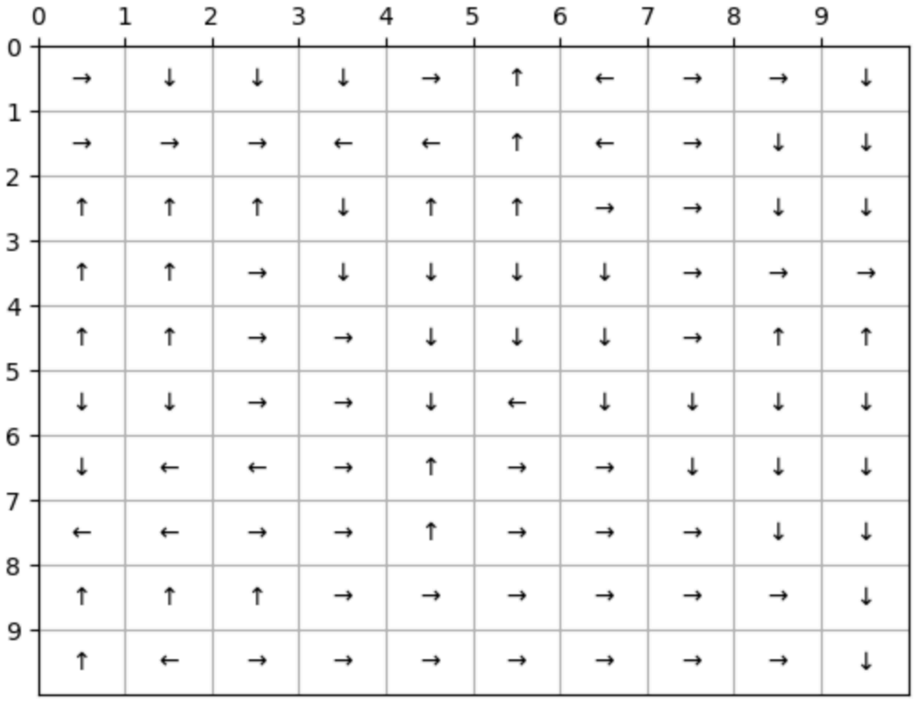
Both of them, as value of state decreases, optimal value decreases, the color of the shade also get lighter. The original reward function exhibits three distinct regions with low rewards, and the extracted reward function demonstrates a similar characteristic, although to a lesser degree.

Differences:

The reward scales are different. From question 14, the scale is from -150 to 50, from question 3, the scale is from -1 to 5

The heat map from question 3 is much more like reward function. But for question 14, the extracted reward function is noisier.

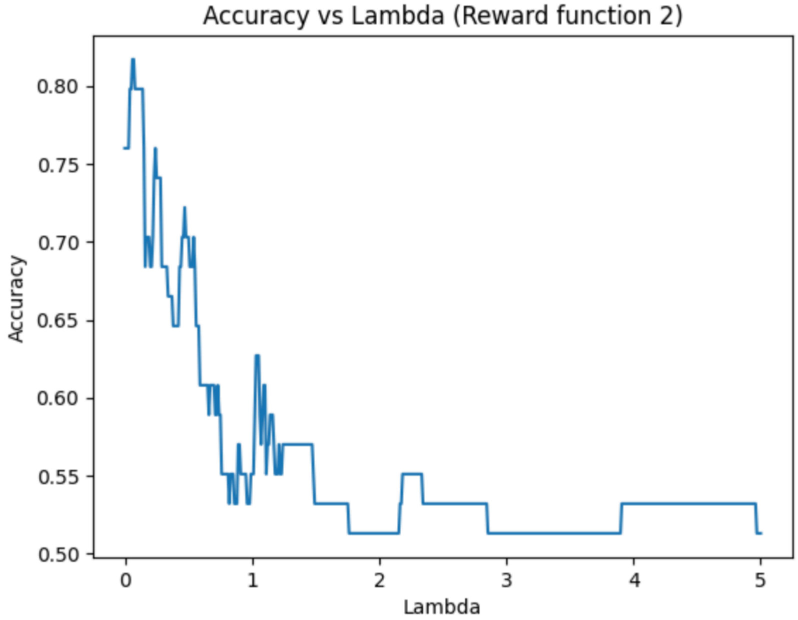
Question 16:



	0	1	2	3	4	5	6	7	8	9
0	→	↓	↓	↓	→	↑	←	→	→	↓
1	→	→	→	←	←	↑	←	→	↓	↓
2	↑	↑	↑	↓	↑	↑	→	→	↓	↓
3	↑	↑	→	↓	↓	↓	↓	→	→	→
4	↑	↑	→	→	↓	↓	↓	→	↑	↑
5	↓	↓	→	→	↓	←	↓	↓	↓	↓
6	↓	←	←	→	↑	→	→	↓	↓	↓
7	←	←	→	→	↑	→	→	→	↓	↓
8	↑	↑	↑	→	→	→	→	→	→	↓
9	↑	←	→	→	→	→	→	→	→	↓

Question 17:
For similarities, they both has a trend to the right down corner where it has the optimal value. For differences, in extracted reward function, there are some states moving out of the grid. There are also in some states that the agent has not the trend towards to the right down corner, because the optimal value for thoes state is higher than certain neighbor, and it follow the higher values.

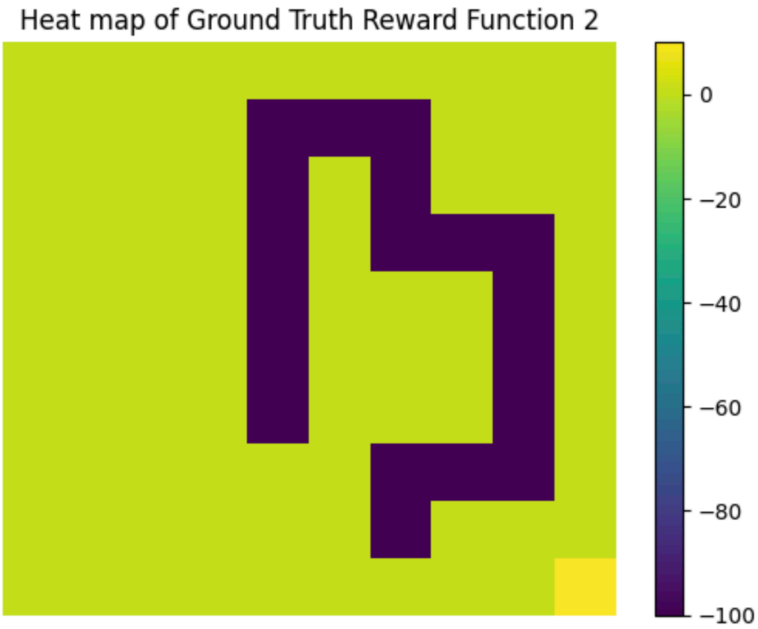
Question 18:

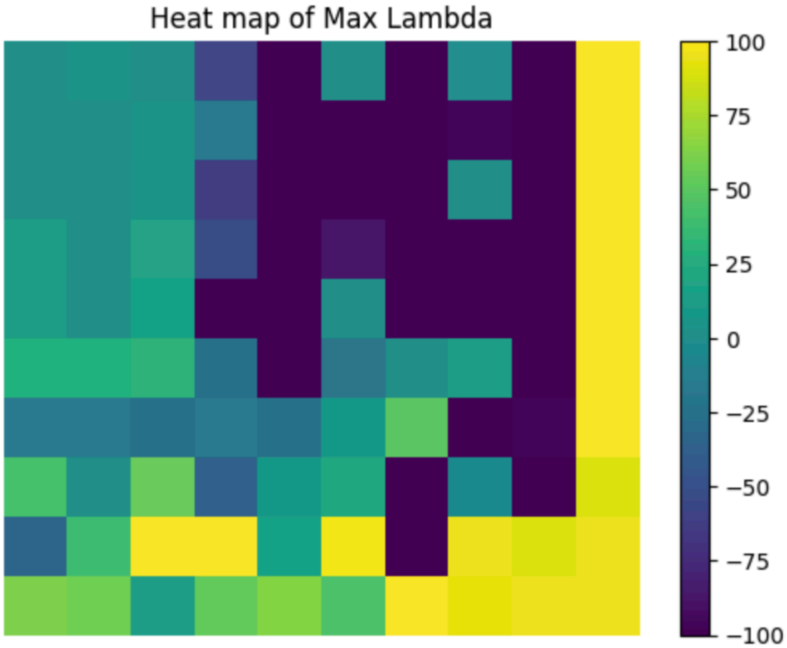


Question 19:

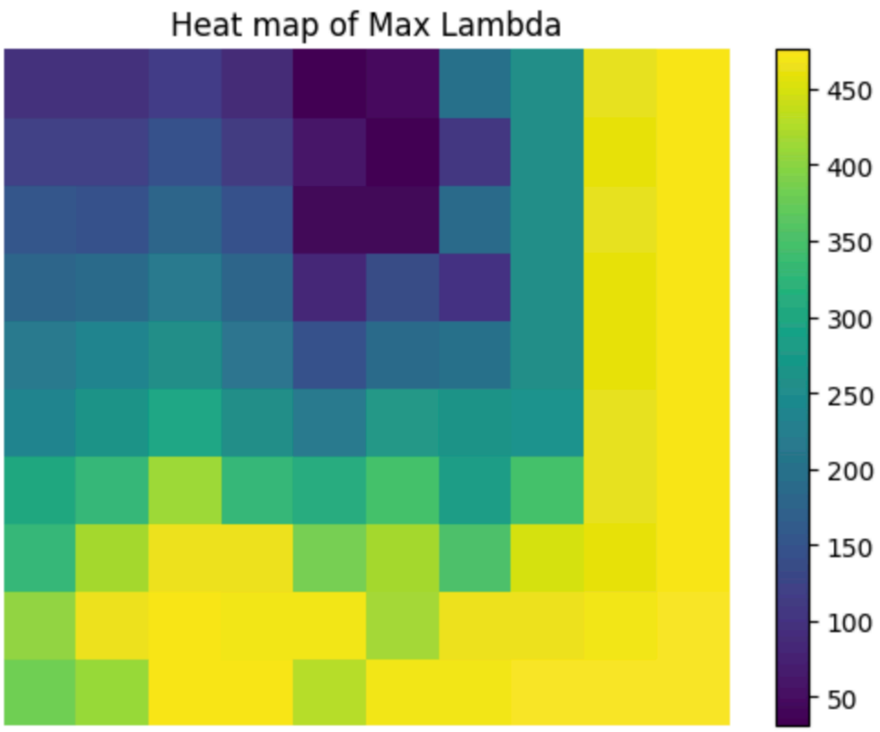
The best lamda $\lambda_{max}^{(2)} = 0.06012024048096192$ with accuracy of 0.817

Question 20:





Question 21:
Heat map of optimal state values



Question 22:
Similarities:

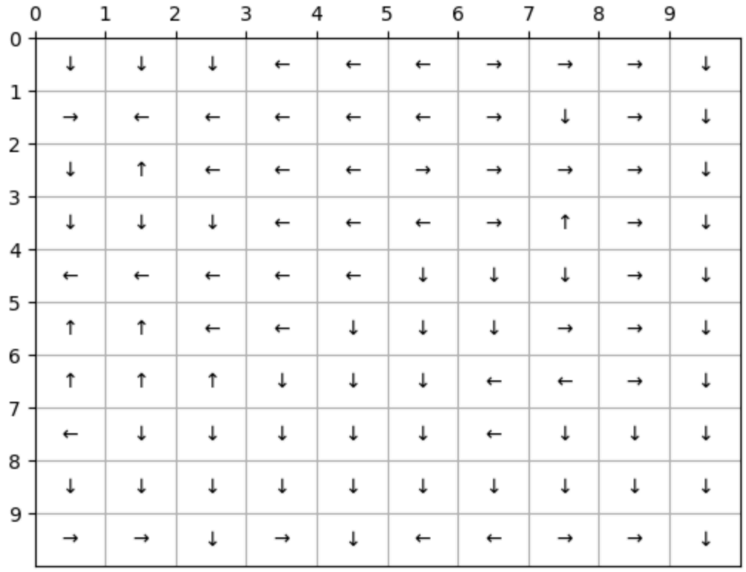
Both has the highest value state at right down conor, state 99.

Differences:

The scales are different. From question 21, scale is from 0-500, but for question 7, scale is -10-50.

In contrast to the original reward function, the extracted reward function features two distinct regions of high-value states: one located in the bottom right corner and another in the bottom left half of the grid.

Question 23:



	0	1	2	3	4	5	6	7	8	9
0	↓	↓	↓	←	←	←	→	→	→	↓
1	→	←	←	←	←	←	→	↓	→	↓
2	↓	↑	←	←	←	→	→	→	→	↓
3	↓	↓	↓	←	←	←	→	↑	→	↓
4	←	←	←	←	←	↓	↓	↓	→	↓
5	↑	↑	←	←	↓	↓	↓	→	→	↓
6	↑	↑	↑	↓	↓	↓	←	←	→	↓
7	←	↓	↓	↓	↓	↓	←	↓	↓	↓
8	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
9	→	→	↓	→	↓	←	←	→	→	↓

Question 24:
For similarities, both of the reward function has a trend to the right down corner where it has the optimal value. Down arrows are maximum and upward arrow facing arrows are minimum. They also avoid the center portion, which the agent tries to move with the lower rewards. There are instances in both reward functions where two states are oriented away from each other.

For differences, in extracted reward function, there are some states moving out of the grid, but they still try to move to the right down corner. For example, at 4th row, last column, it goes out of the grid, but in question 23, it goes all the way down to the state 99. The extracted reward function also exhibits a few states that are directed towards each other.

Question 25:
The first discrepancy happens when the states located at the boundaries exhibit estimated optimal policies that involve moves leading them off the grid. The second discrepancy happens on the initial position of the agent within the grid, there is a possibility of getting trapped in local optimum. To solve these two discrepancies, we decrease ϵ to 0.0000001 and decrease discount to 0.55 in order to make the convergence criterion stricter.

	0	1	2	3	4	5	6	7	8	9
0	↓	↓	↓	←	←	→	→	↑	→	↓
1	↓	↓	↓	←	←	→	→	↑	→	↓
2	↓	↓	↓	←	←	→	→	↑	→	↓
3	↓	↓	←	←	←	←	→	↑	→	↓
4	←	←	←	←	←	↓	↓	↓	→	↓
5	↑	↑	←	←	←	↓	↓	→	→	↓
6	↑	↑	↑	↓	→	↓	←	↓	→	↓
7	↓	↓	↓	↓	↓	↓	←	↓	↓	↓
8	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
9	↓	←	←	←	←	←	→	→	→	↓