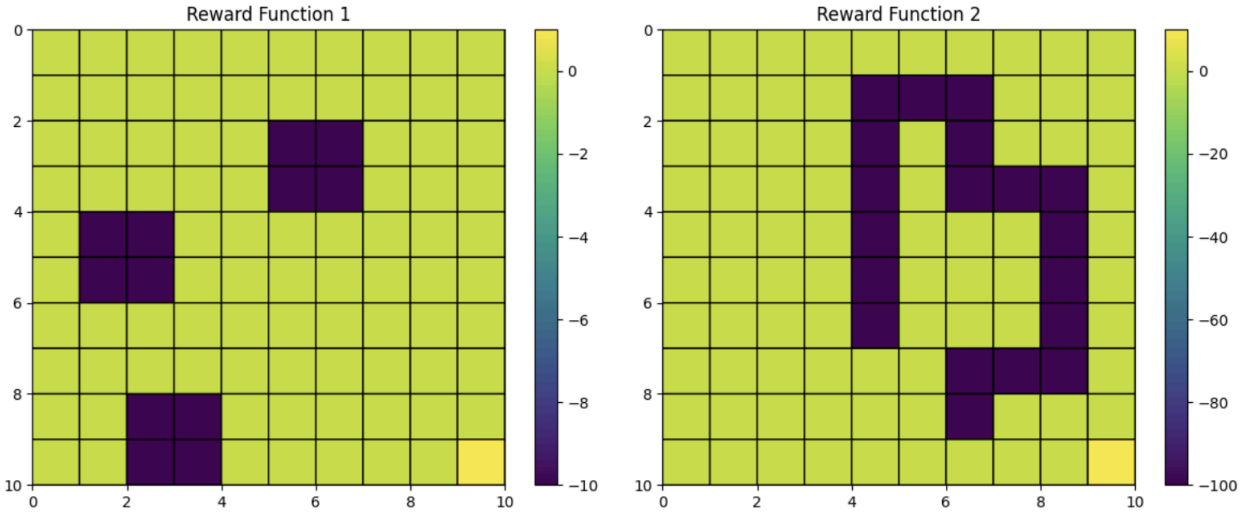
Reinforcement learning and Inverse Reinforcement learning

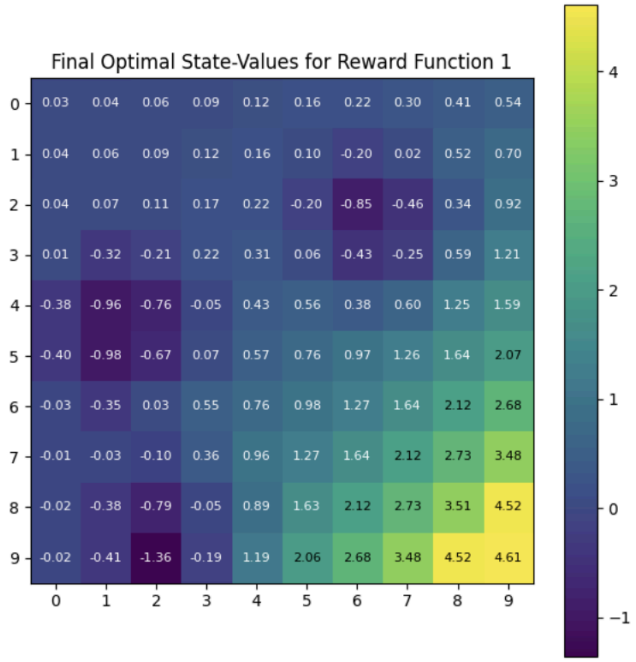
Qikai Feng 106395231
Yi Han 606335602
Xinxin Chang 306405941

Question 1



Question 2

To compute the optimal state values for Reward Function 1, we first built a grid-based MDP environment consisting of a 10x10 state space, four possible actions (`up`, `down`, `left`, `right`), a transition noise parameter `w=0.1`, and a discount factor of γ`=0.8`. The reward structure includes a goal state at (9,9) with a reward of +1 and twelve trap states with a penalty of -10 (at coordinates shown in Figure 6). Using the value iteration algorithm, we initialized state-values to 0 and iteratively applied the Bellman optimality update until convergence with a threshold of ε`=0.01`. The final value function result is visualized as a heat map in Figure below. The plot shows that values gradually increase from the upper-left region to the goal in the lower-right, while trap regions around the middle and lower-left severely depress value estimates. This reflects the agent's learned avoidance of negative-reward states and preference for safe paths toward the goal.



The value iteration algorithm converged in 22 steps. To visualize the learning progression during value iteration, we plotted the value function at five linearly spaced steps: Step 1, Step 6, Step 11, Step 16, and Step 22 (the final step before convergence). Early in the process (Step 1), all state values are nearly zero, except for states near traps which start to reflect large penalties due to immediate negative rewards. As iterations proceed, value information propagates outward from both the goal and trap regions. By Step 11, we can observe a clearer gradient structure forming across the grid, with increasing values as states get closer to the goal and decreasing values near traps. At convergence (Step 22), the state values stabilize, reflecting an optimal policy where the agent maximizes cumulative reward by balancing risk and reward. Notably, the final value map shows a distinct avoidance path around traps and higher confidence toward reaching the goal along safe regions.

State-values at Step 1

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	-0.33	-0.33	0.00	0.00
0.00	0.00	0.00	0.00	0.00	-0.33	-0.67	-0.67	-0.33	0.00
0.00	-0.33	-0.33	0.00	0.00	-0.33	-0.67	-0.67	-0.33	0.00
-0.33	-0.67	-0.67	-0.33	0.00	0.00	-0.33	-0.33	0.00	0.00
-0.33	-0.67	-0.67	-0.33	0.00	0.00	0.00	0.00	0.00	0.00
0.00	-0.33	-0.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	-0.33	-0.33	0.00	0.00	0.00	0.00	0.00	0.00
0.00	-0.33	-0.67	-0.67	-0.33	0.00	0.00	0.00	0.00	0.90
0.00	-0.33	-1.00	-1.00	-0.33	0.00	0.00	0.00	0.90	0.93

State-values at Step 6

-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.01	-0.01	-0.00	-0.00
-0.00	-0.00	-0.00	-0.00	-0.00	-0.02	-0.38	-0.38	-0.02	-0.00
-0.00	-0.01	-0.01	-0.00	-0.01	-0.38	-1.00	-1.00	-0.38	-0.01
-0.02	-0.38	-0.38	-0.02	-0.02	-0.38	-1.00	-1.00	-0.38	0.16
-0.40	-1.00	-1.00	-0.38	-0.02	-0.02	-0.38	-0.38	0.19	0.45
-0.40	-1.01	-1.00	-0.38	-0.01	-0.00	-0.01	0.20	0.49	0.91
-0.02	-0.39	-0.40	-0.03	-0.00	-0.00	0.21	0.50	0.95	1.51
-0.00	-0.03	-0.41	-0.39	-0.02	0.21	0.50	0.95	1.55	2.31
-0.01	-0.38	-1.02	-1.01	-0.17	0.48	0.95	1.55	2.33	3.34
-0.02	-0.41	-1.39	-1.21	0.06	0.89	1.51	2.31	3.34	3.43

State-values at Step 11

-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.01	0.02	0.09	0.20
-0.00	-0.00	-0.00	-0.00	-0.00	-0.02	-0.38	-0.29	0.18	0.35
-0.00	-0.01	-0.01	-0.00	-0.01	-0.38	-1.00	-0.79	-0.02	0.56
-0.02	-0.38	-0.38	-0.02	0.02	-0.27	-0.77	-0.60	0.23	0.85
-0.40	-1.00	-1.00	-0.34	0.10	0.22	0.02	0.24	0.88	1.23
-0.40	-1.01	-0.96	-0.26	0.23	0.40	0.61	0.90	1.28	1.70
-0.03	-0.39	-0.29	0.21	0.40	0.62	0.91	1.28	1.75	2.32
-0.01	-0.04	-0.40	0.01	0.60	0.91	1.28	1.75	2.36	3.12
-0.02	-0.38	-1.00	-0.40	0.53	1.27	1.75	2.36	3.15	4.15
-0.02	-0.41	-1.37	-0.54	0.83	1.69	2.32	3.12	4.15	4.24

State-values at Step 16

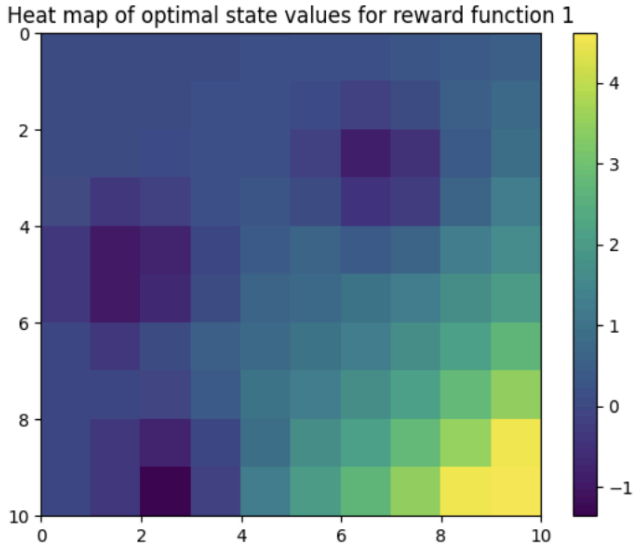
-0.00	-0.00	0.00	0.01	0.04	0.07	0.12	0.21	0.32	0.44
-0.00	0.00	0.01	0.04	0.07	0.02	-0.29	-0.07	0.42	0.60
-0.00	-0.00	0.03	0.08	0.13	-0.29	-0.93	-0.55	0.24	0.82
-0.02	-0.38	-0.30	0.12	0.22	-0.04	-0.52	-0.34	0.50	1.11
-0.40	-1.00	-0.85	-0.14	0.33	0.47	0.28	0.51	1.15	1.49
-0.40	-1.00	-0.76	-0.03	0.48	0.66	0.88	1.16	1.54	1.97
-0.03	-0.38	-0.07	0.45	0.66	0.89	1.18	1.55	2.02	2.59
-0.01	-0.03	-0.19	0.27	0.87	1.17	1.55	2.02	2.63	3.39
-0.02	-0.38	-0.88	-0.15	0.79	1.53	2.02	2.63	3.42	4.42
-0.02	-0.41	-1.36	-0.29	1.09	1.96	2.59	3.39	4.42	4.51

State-values at Step 22

0.03	0.04	0.06	0.09	0.12	0.16	0.22	0.30	0.41	0.54
0.04	0.06	0.09	0.12	0.16	0.10	-0.20	0.02	0.52	0.70
0.04	0.07	0.11	0.17	0.22	-0.20	-0.85	-0.46	0.34	0.92
0.01	-0.32	-0.21	0.22	0.31	0.06	-0.43	-0.25	0.59	1.21
-0.38	-0.96	-0.76	-0.05	0.43	0.56	0.38	0.60	1.25	1.59
-0.40	-0.98	-0.67	0.07	0.57	0.76	0.97	1.26	1.64	2.07
-0.03	-0.35	0.03	0.55	0.76	0.98	1.27	1.64	2.12	2.68
-0.01	-0.03	-0.10	0.36	0.96	1.27	1.64	2.12	2.73	3.48
-0.02	-0.38	-0.79	-0.05	0.89	1.63	2.12	2.73	3.51	4.52
-0.02	-0.41	-1.36	-0.19	1.19	2.06	2.68	3.48	4.52	4.61

Question 3

To provide a visual interpretation of the agent's learned state values across the 2D grid, we generated a heat map using `plt.pcolor` as suggested in the hint. The heat map represents the final optimal state values computed from value iteration with Reward Function 1. In the figure, brighter (yellow) areas indicate higher state values, typically found near the goal state at (9,9), while darker (purple) regions correspond to states close to traps or less optimal paths. This visualization clearly highlights how the value function propagates from the goal state backwards through the grid, and how the negative rewards from traps suppress value propagation around those regions. The plot also aligns with the numerical results in Question 2 and confirms that the learned policy would guide the agent along high-value paths while avoiding low-value (risky) areas.



Question 4

The distribution of the optimal state values across the 2D grid is directly influenced by the reward function and the agent's ability to reach high-reward states while avoiding negative ones. From the heat map generated in Question 3, we observe that the values gradually increase as we move toward the bottom-right corner of the grid (state (9,9)), which contains the goal reward of +1.0. This region has the highest values because it is the most desirable state, and the surrounding states also inherit high values due to the discounted future reward propagating backward through the grid. Conversely, states near traps—such as (2,6), (2,7), (3,6), and others—have significantly lower values due to the large negative reward of -10. These low-value zones also affect adjacent states, causing their values to drop because of the possibility of transitioning into traps due to the stochastic movement.

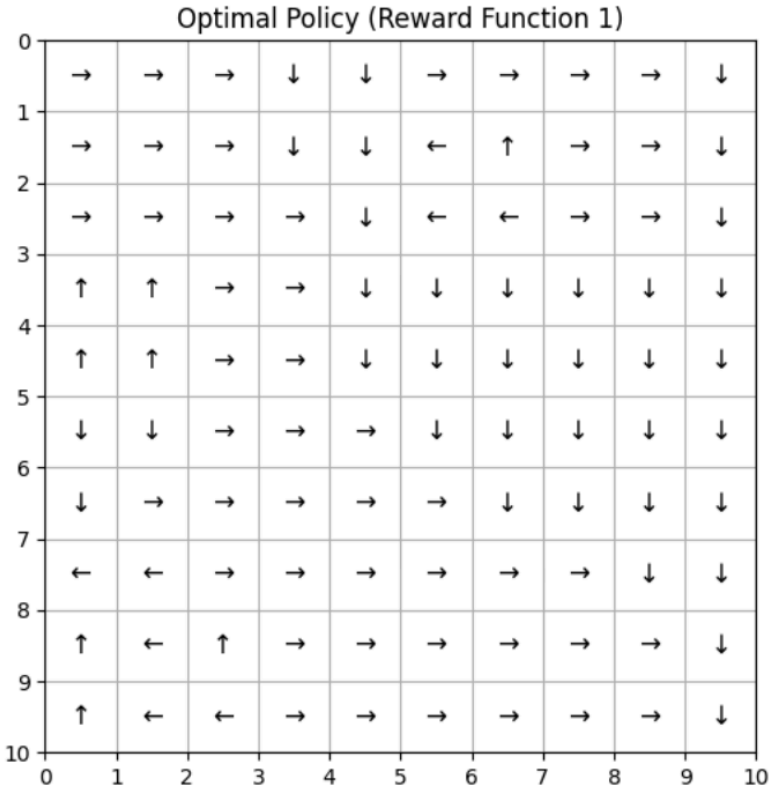
The value distribution forms a gradient: values are high near the goal and diminish further away, especially when potential paths are blocked or disrupted by traps. The agent learns to avoid these low-value zones and prefers safer, higher-valued paths. This explains the non-uniform, yet structured, spread of values across the grid—reflecting both optimality and caution.

Question 5

To compute the optimal policy for Reward Function 1, I implemented the policy extraction step by evaluating the expected return for each possible action at every state using the final state values obtained from value iteration. The optimal action was selected as the one yielding the highest expected return, and the resulting policy was visualized using directional arrows (↑, ↓, ←, →) on a 10×10 grid. Each arrow indicates the direction the agent should move to maximize its long-term reward from that state.

The resulting policy plot shows that the agent generally avoids trap states (with -10 rewards) and navigates toward the goal state at (9, 9), which provides a reward of +1. This matches my intuition: the arrows bend around trap zones and converge toward the bottom-right corner, forming an efficient path while minimizing negative rewards.

Yes, it is possible for the agent to infer the optimal action by comparing the optimal state values of its neighboring states. Since value iteration propagates value information backward from the goal and updates each state based on its neighbors, the direction with the highest neighboring value often corresponds to the best action. Thus, a greedy policy based on value comparisons can approximate the optimal policy effectively.



Question 6

In this question, we replaced Reward Function 1 with Reward Function 2 and reran the value iteration algorithm using the same convergence threshold $\epsilon = 0.01$\epsilon = 0.01$\epsilon=0.01$. The new reward configuration significantly altered the landscape of state values. As shown in the printed value grid, the optimal state values now vary much more dramatically across the grid, with extremely negative values clustering near the penalty regions and much higher positive values accumulating near the goal at the bottom-right corner. The value iteration algorithm converged in 32 steps, slightly more than under Reward Function 1. This is expected, as the more complex reward landscape introduced by Reward Function 2—especially the inclusion of multiple large

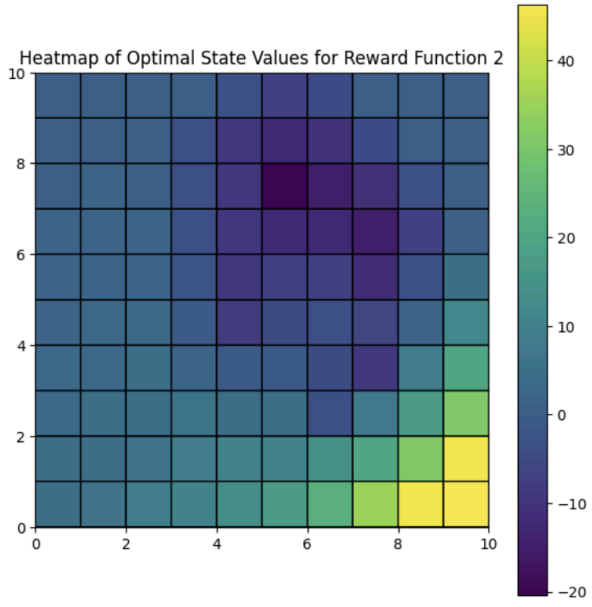
negative traps—requires more iterations for the value function to stabilize across all states.

State-values for Reward Function 2

0.52	0.62	0.50	0.17	-3.75	-6.66	-4.07	-0.23	-0.01	-0.00
0.67	0.81	0.65	-3.21	-9.68	-12.55	-10.44	-4.05	-0.22	-0.01
0.87	1.06	0.99	-3.05	-9.93	-20.39	-15.29	-10.35	-3.84	-0.03
1.12	1.38	1.43	-2.70	-9.47	-13.02	-12.69	-14.97	-7.19	0.11
1.44	1.79	2.01	-2.25	-8.96	-7.24	-6.75	-11.98	-3.50	5.06
1.85	2.32	2.77	-1.60	-8.19	-4.35	-3.55	-6.43	1.63	11.60
2.37	3.00	3.75	1.17	-0.34	-0.69	-4.28	-9.51	8.03	20.08
3.04	3.86	4.96	6.19	4.45	4.01	-3.32	7.65	17.25	31.05
3.89	4.97	6.39	8.22	10.46	10.19	14.01	19.73	31.49	45.24
4.89	6.30	8.16	10.57	13.69	17.74	23.11	34.70	45.34	46.35

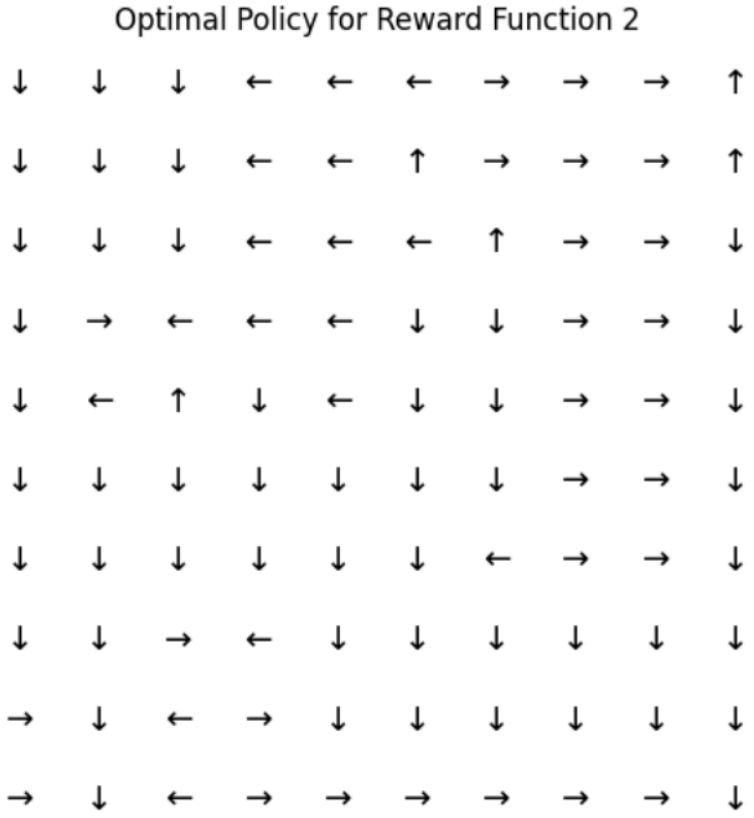
Question 7

The heat map generated for Reward Function 2 reveals a clear pattern of value propagation from the terminal reward at (9,9), with steep value gradients radiating outward. States closer to the goal accumulate high positive values, whereas states near the trap zones (especially in the middle-upper and left-middle areas) exhibit strong negative values, forming dark “basins” on the map. The values decrease sharply as states get closer to these penalty regions due to the -20 reward, which significantly lowers the expected return of neighboring paths. Overall, the heat map visually confirms that the agent learns to avoid high-risk areas and favor paths leading safely toward the goal. This distribution matches our expectations and reflects the agent's rational adaptation to the new reward structure.



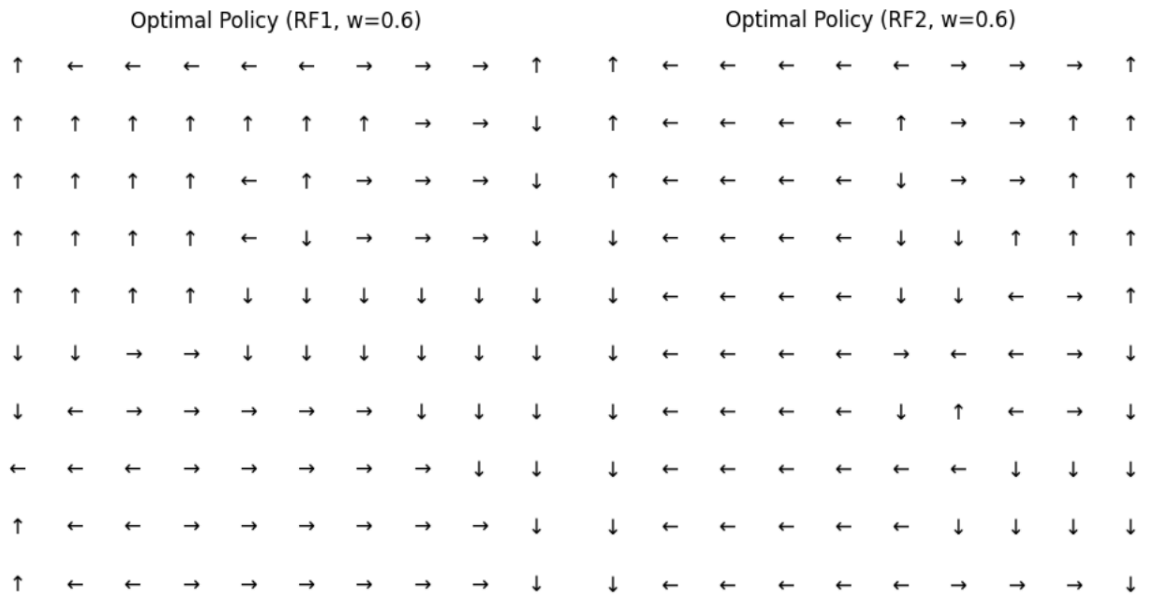
Question 8

To compute the optimal policy for Reward Function 2, I used the final state-value function from the previous step and determined the best action at each state by maximizing the expected value over all possible next states. The resulting policy is visualized as a grid of arrows, where each arrow indicates the direction the agent should move to maximize cumulative reward. The policy closely aligns with intuition: the agent consistently avoids low-value trap zones and moves steadily toward the high-reward goal at the bottom-right corner. In many states, the agent appears to follow gradients of increasing value, which confirms that it is indeed possible to infer optimal actions by comparing the values of neighboring states. This policy reflects the agent's learned behavior of navigating efficiently while minimizing risk.



Question 9

After increasing the hyperparameter www from 0.1 to 0.6, the optimal policy maps for both reward functions show noticeable changes. In the policy maps with w=0.6, the arrows indicate more decisive and consistent movement toward the goal region. For RF1, the agent aggressively navigates down and right, avoiding uncertain or low-value areas. For RF2, which includes more traps or negative regions, the policy becomes more conservative in the middle but quickly moves down-right once it escapes the penalizing zones. Compared to the policies under w=0.1, which were more cautious and dispersed, the policies under w=0.6 are more focused and goal-directed.



In terms of state values, increasing www amplifies the value gradients across the grid. Under w=0.1, the values for both reward functions rise gradually and remain relatively moderate, suggesting a more balanced trade-off between short-term rewards and long-term goals. However, under w=0.6, state values vary more dramatically. In RF1, the goal state and its neighboring cells are assigned significantly higher values, and the penalty for deviating is more severe. In RF2, the effect is even more extreme: states near penalty zones have large negative values, and the high-value path becomes narrower. This suggests that a higher www places more emphasis on long-term rewards and punishes suboptimal detours more heavily.

Overall, increasing w to 0.6 produces sharper, more goal-oriented policies and highlights risk zones more clearly. This aligns well with our project's need for efficient and reliable pathfinding. Therefore, we choose w=0.6 for the subsequent stages.

Optimal state-values for Reward Function 1 (w=0.1)

0.04	0.05	0.08	0.11	0.15	0.21	0.28	0.37	0.49	0.61
0.02	0.04	0.06	0.08	0.10	-0.11	0.09	0.47	0.63	0.79
0.01	0.02	0.03	0.05	-0.19	-0.60	-0.26	0.36	0.81	1.02
-0.01	-0.26	-0.23	0.05	0.08	-0.25	-0.10	0.54	1.05	1.32
-0.28	-0.73	-0.47	0.09	0.47	0.36	0.55	1.04	1.35	1.70
-0.26	-0.63	-0.37	0.22	0.63	0.81	1.05	1.35	1.73	2.18
0.03	-0.12	0.19	0.62	0.82	1.05	1.35	1.73	2.22	2.81
0.06	0.09	0.14	0.54	1.04	1.35	1.73	2.22	2.84	3.61
0.04	-0.20	-0.42	0.30	1.08	1.73	2.22	2.84	3.63	4.63
0.01	-0.28	-0.98	0.28	1.41	2.18	2.81	3.61	4.63	4.70

Optimal state-values for Reward Function 1 (w=0.6)

-0.02	-0.03	-0.04	-0.09	-0.24	-0.61	-0.60	-0.23	-0.07	-0.03
-0.05	-0.09	-0.13	-0.24	-0.86	-2.96	-2.94	-0.80	-0.17	-0.04
-0.23	-0.49	-0.57	-0.73	-3.08	-6.14	-6.07	-2.84	-0.42	-0.02
-1.01	-2.95	-3.04	-1.52	-3.47	-6.20	-6.04	-2.79	-0.34	0.09
-3.58	-6.27	-6.28	-3.49	-1.55	-2.99	-2.80	-0.62	0.08	0.29
-3.81	-6.55	-6.40	-3.21	-0.79	-0.50	-0.31	0.11	0.38	0.54
-1.46	-3.58	-3.78	-1.39	-0.37	-0.02	0.18	0.41	0.67	0.92
-0.91	-1.72	-3.93	-3.25	-0.75	0.07	0.40	0.69	1.07	1.53
-1.10	-3.51	-6.91	-6.54	-2.82	-0.08	0.60	1.06	1.67	2.53
-1.28	-4.09	-9.32	-8.95	-3.35	-0.06	0.82	1.51	2.53	2.95

Optimal state-values for Reward Function 2 (w=0.1)

0.65	0.79	0.82	0.53	-2.39	-4.24	-1.92	1.13	1.59	2.03
0.83	1.02	1.06	-1.88	-6.75	-8.68	-6.37	-1.30	1.92	2.61
1.06	1.31	1.45	-1.64	-6.76	-13.92	-9.65	-5.51	-0.13	3.36
1.36	1.69	1.94	-1.24	-6.34	-7.98	-7.95	-9.43	-1.92	4.39
1.73	2.17	2.59	-0.74	-5.85	-3.26	-3.24	-7.43	1.72	9.16
2.21	2.78	3.41	-0.04	-5.11	-0.55	-0.49	-2.98	6.58	15.35
2.82	3.55	4.48	3.02	2.48	2.88	-0.47	-4.91	12.69	23.30
3.58	4.54	5.79	7.29	6.72	7.24	0.93	12.37	21.16	33.48
4.56	5.79	7.40	9.44	12.01	12.89	17.10	23.01	33.78	46.53
5.73	7.32	9.39	12.04	15.45	19.82	25.50	36.16	46.58	47.31

Optimal state-values for Reward Function 2 (w=0.6)

-0.24	-0.58	-2.46	-10.74	-38.95	-53.71	-41.26	-13.48	-4.82	-2.62
-0.30	-0.95	-5.13	-30.82	-67.58	-84.80	-74.42	-40.00	-12.29	-4.83
-0.35	-1.13	-6.02	-34.70	-76.42	-117.75	-97.50	-75.37	-39.97	-13.57
-0.38	-1.18	-6.15	-34.89	-74.31	-94.80	-94.01	-94.05	-74.12	-41.83
-0.35	-1.14	-6.02	-34.17	-70.12	-69.26	-67.39	-89.34	-85.29	-57.94
-0.29	-1.04	-5.65	-32.71	-65.44	-59.98	-50.18	-68.14	-84.89	-61.24
-0.20	-0.80	-4.51	-27.31	-45.30	-56.19	-62.76	-84.62	-79.41	-49.44
-0.10	-0.36	-1.73	-8.01	-30.75	-47.79	-73.22	-77.53	-60.96	-26.43
-0.05	-0.14	-0.58	-2.37	-8.81	-31.07	-44.68	-48.36	-20.35	11.35
-0.03	-0.07	-0.22	-0.83	-2.87	-9.19	-25.81	-2.83	14.57	22.97

Question 10

Let$|S|$be the number of states, A the action set, a_1 the expert action, γ the discount factor, λ the penalty weight, and R_{max} the maximum absolute reward.

1. Decision vector $x \in R^{3|S|}$:

$$x = \begin{bmatrix} R(1) \\ \vdots \\ R(|S|) \\ t_1 \\ \vdots \\ t_{|S|} \\ u_1 \\ \vdots \\ u_{|S|} \end{bmatrix} \in \mathbb{R}^{3|S|}$$

2. Objective coefficient vector c: maximize $\sum_i (t_i - \lambda u_i)$ is equivalent to minimizing $-c^T x$, where

$$c = \left[\underbrace{0, \ldots, 0}_{|S|}, \underbrace{1, \ldots, 1}_{|S|}, \underbrace{-\lambda, \ldots, -\lambda}_{|S|} \right]^T$$

3. Inequality constraints $Dx \leq b$, total rows $m = (|A| - 1)|S| + 4|S|$, split into:
   a) Core IRL constraints
   For each state i and each non-expert action $a \neq a_1$:

$$(P_{a_1}(i,:) - P_a(i,:))\,(I - \gamma P_{a_1})^{-1} R \;\geq\; t_i \iff \left[(P_{a_1} - P_a)(I - \gamma P_{a_1})^{-1}\right]_{i,:} R - e_i^T t \;\leq\; 0$$

   b) Pairwise bounds

$$-u_i \leq R_i \leq u_i \quad \Longrightarrow \quad \begin{cases} R_i - u_i \leq 0, \\ -R_i - u_i \leq 0 \end{cases}$$

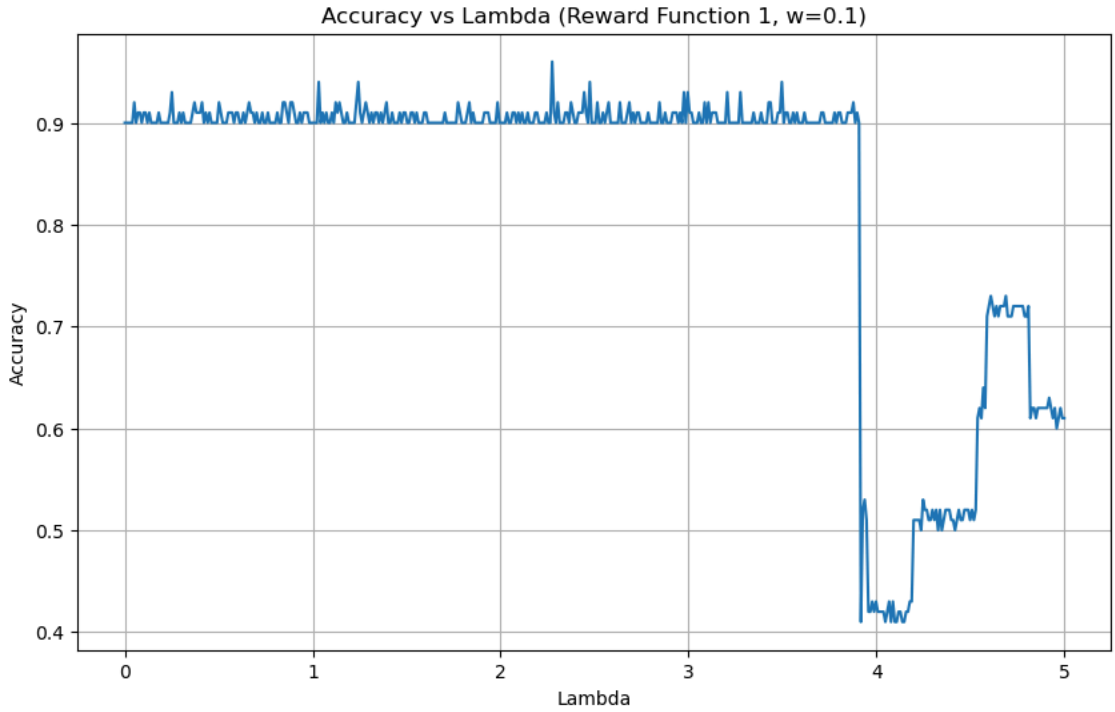
   c) Absolute bound

$$|R_i| \leq R_{\max} \quad \Longrightarrow \quad \begin{cases} R_i \leq R_{\max}, \\ -R_i \leq R_{\max} \end{cases}$$

4. Right-hand side vector b:

$$b = \underbrace{0, \ldots, 0}_{(|A|-1)\,|S| + 2\,|S|}, \underbrace{R_{\max}, \ldots, R_{\max}}_{2\,|S|}{}^T$$

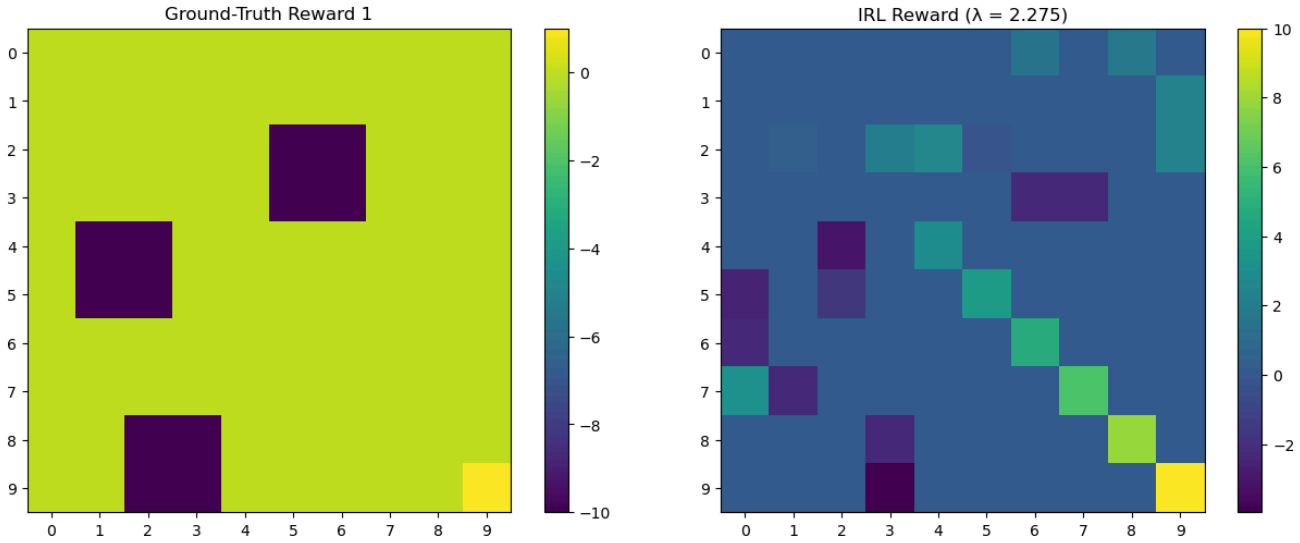
Question 11



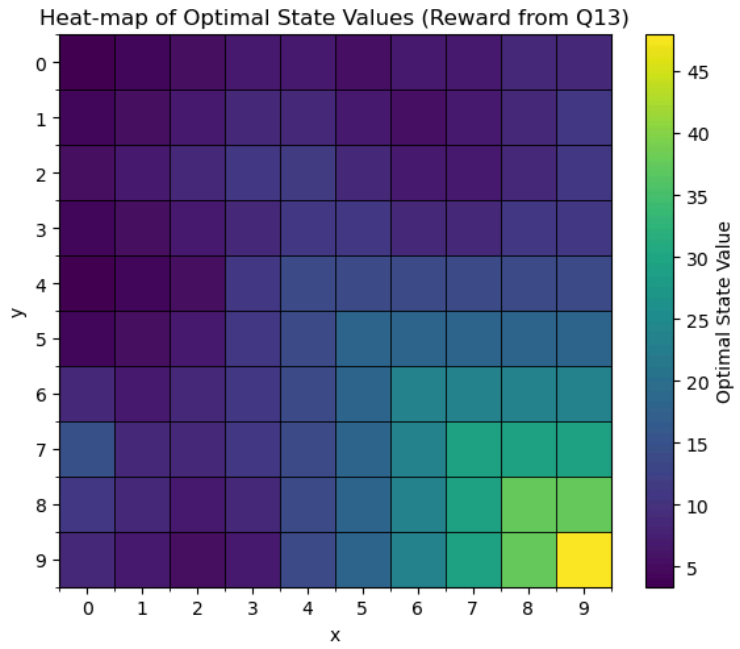
Question 12

The best value of accuracy is 0.96 when $\lambda = 2.2745$.

Question 13



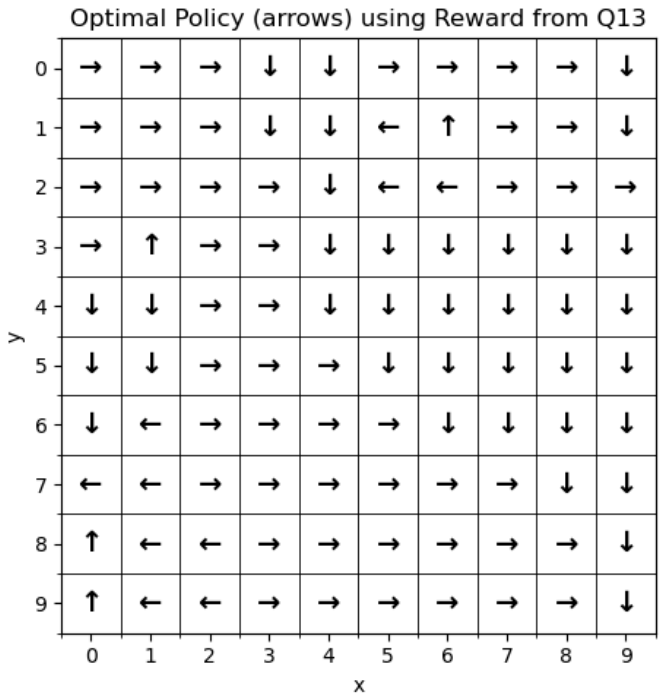
Question 14



Question 15

Comparing the heatmaps from Question 3 and Question 14, both plots show the highest optimal state values in the bottom-right corner, corresponding to the positive goal reward, and in both cases, state values generally decrease as the distance from the goal increases. However, a significant difference is the absence of distinct negative value regions (darker dips) in the Question 14 heatmap, which were clearly present in the Question 3 heatmap corresponding to the negative reward trap states. The Question 14 heatmap appears smoother in its value distribution with less pronounced local variations, indicating that the extracted reward function, while guiding the policy towards the goal, did not fully recover the specific negative reward values of the trap states.

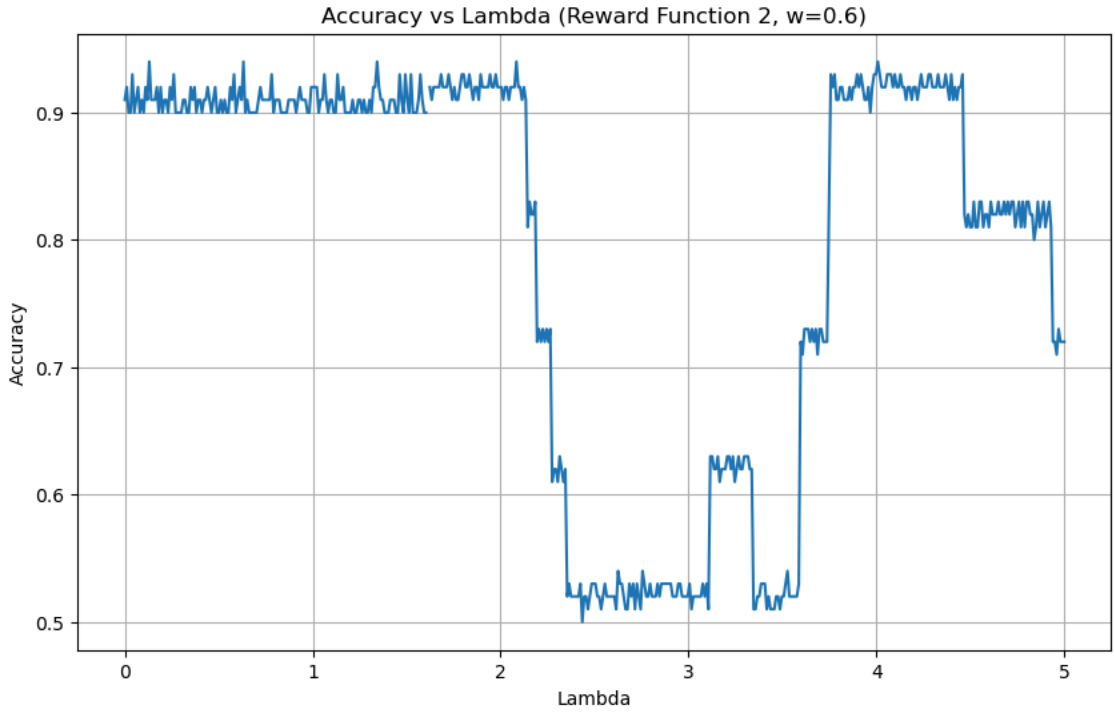
Question 16



Question 17

Comparing the policy figures from Question 5 and Question 16, both maps fundamentally direct the agent toward the goal state in the bottom-right corner. The overall flow of actions in both figures guides the agent from various starting positions towards this objective. However, a crucial distinction emerges in their detailed execution, particularly around the negative reward trap regions. The optimal policy derived from the ground truth reward (Question 5) exhibits meticulous navigation around these undesirable states, showing clear repulsive actions or specific detours to avoid the penalties. In stark contrast, the policy derived from the extracted reward (Question 16) presents a smoother, more uniformly directed path towards the goal, notably lacking these pronounced avoidance maneuvers around the traps. This difference underscores a characteristic of Inverse Reinforcement Learning algorithms using simplified Linear Programming (LP) formulations, which aim to extract a reward function that adequately explains the expert's observed behavior, rather than perfectly reconstructing the underlying ground truth reward. If the expert policy consistently avoids certain regions, the LP, with its objective to make the expert policy optimal and potentially with regularization on reward magnitudes, may find a simpler reward landscape (e.g., a smooth gradient towards the goal) to be sufficient to incentivize the observed optimal actions, thereby not explicitly recovering the specific negative trap values of the original reward function.

Question 18

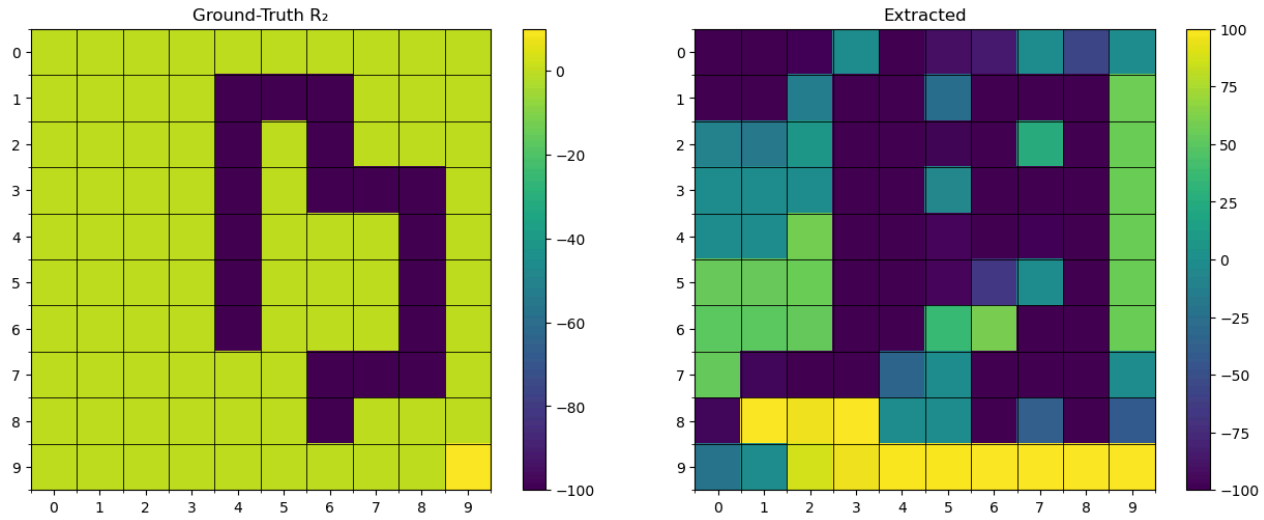


Question 19

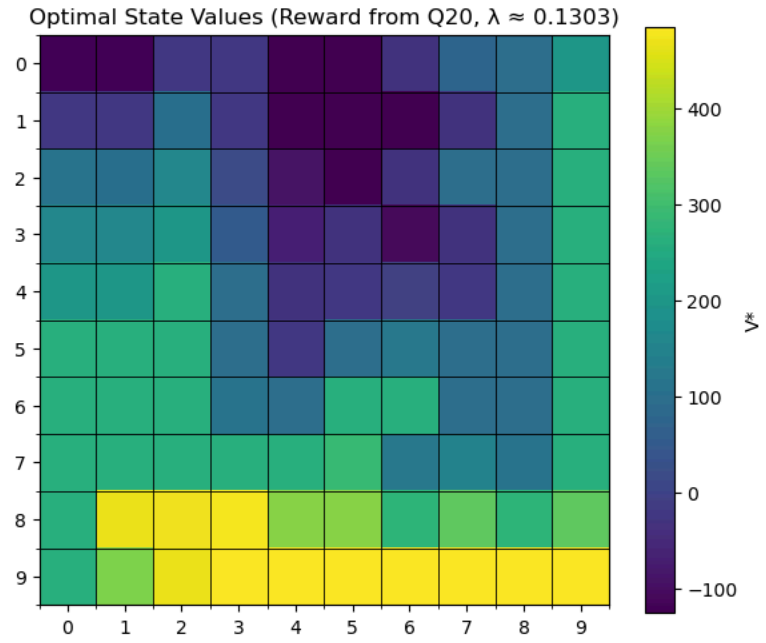
```
Lambda_max for Reward Function 2 (w=0.6): 0.1303
Maximum Accuracy: 0.9400
```

The best value of accuracy is 0.94 when $\lambda = 0.1303$.

Question 20



Question 21



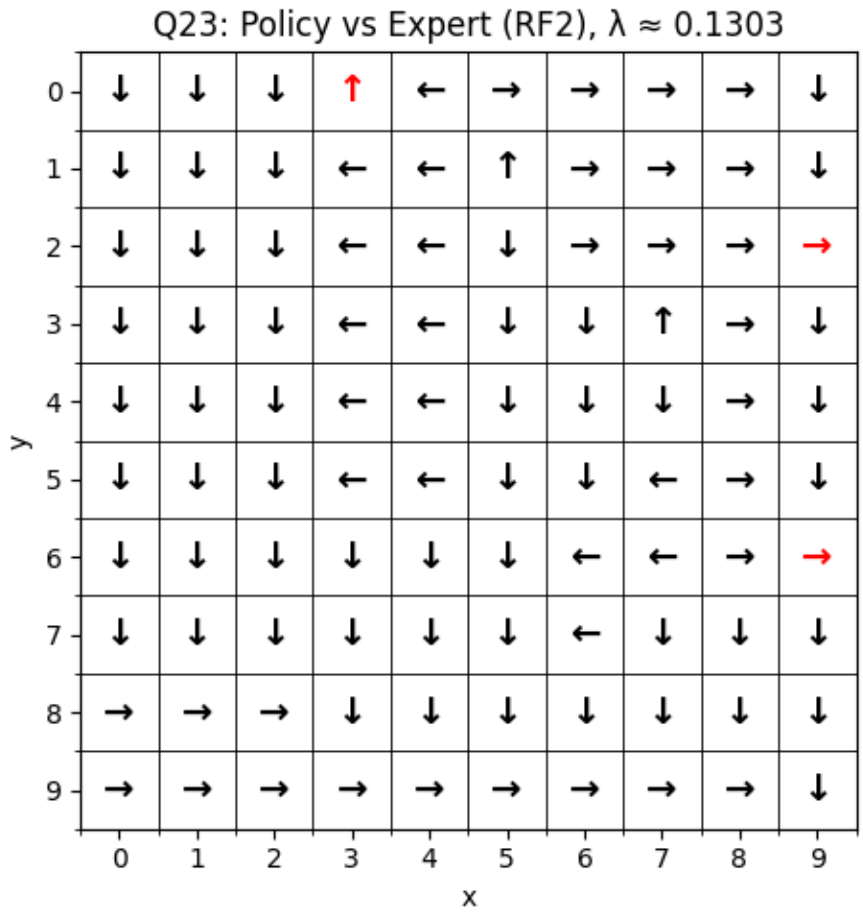
Question 22

The heat map from Question 7 shows the true optimal state values based on Reward Function 2, while the Question 21 heat map shows values derived from the reward function learned via IRL. Both maps successfully identify the large, low-value penalty zone in the upper-middle grid, indicating the IRL process captured the expert's strong aversion to these areas defined by

Reward Function 2's large negative rewards. This shows the IRL's ability to learn key negative constraints from observed behavior.

However, significant differences exist. Firstly, the absolute scale of state values in the IRL-derived map (Q21) is much larger (up to ~450) compared to the ground truth map (Q7, up to ~40). This scaling difference is common in IRL, as the algorithm seeks a reward function that explains the policy, not necessarily one identical in magnitude to the true rewards; the λ parameter and R_{max} constraints in the LP formulation also influence this. More critically, the distribution of high values differs: Q7 shows a focused high-value area at the specific goal (9,9), while Q21 displays the entire bottom row as highly valuable. This suggests the IRL-extracted reward, while guiding the agent towards the correct general area, is less specific about the precise goal state. It may have found that making the entire bottom edge attractive sufficiently explains the expert's policy of moving towards that region, a common outcome where IRL produces a more generalized or ambiguous reward structure than the ground truth. This highlights that multiple reward functions can rationalize the same observed policy, and the one recovered by IRL might not perfectly mirror the original, leading to such variations in the resulting state value landscape.

Question 23



We simply flattened the IRL-extracted reward from Q20 and fed it into our Question 9 value-iteration routine to get the optimal action π∗π^*π∗. The resulting 10 × 10 arrow map (shown above) directs the agent toward the bottom-right high-reward region, with only a handful of tie-break deviations at cells like (0,3), (2,9) and (6,9).

Question 24

Comparison of Policy Maps (Question 9 vs. Question 23)This compares the optimal policy from Question 9, derived from the true Reward Function 2 with w=0.6, and the optimal policy from Question 23, derived from the IRL-extracted reward function, also with w=0.6.
Regarding similarities, both policies effectively guide the agent towards the main reward area defined in Reward Function 2, which is typically the bottom-right of the grid. Furthermore, both policies demonstrate a clear ability to navigate around the significant negative reward areas, or traps, that are part of Reward Function 2. The w=0.6 setting contributes to this avoidance behavior being quite decisive in both cases.
However, there are notable differences. A key distinction lies in the precision of reaching the goal state. The Q9 policy, based on the true reward, is generally more precise in guiding the agent directly to the specific high-reward states, like state 99, with actions near the goal being

typically direct. In contrast, the Q23 policy, derived from the IRL reward, can be less precise. A significant issue, visible in image_47dc33.png (the Q23 policy map), is that policy arrows in the last row (y=9) often still point "down." This is physically impossible and indicates a problem with how the IRL-extracted reward function translates to behavior at boundaries or terminal states. It suggests the agent believes there's value in moving further down, likely because the extracted reward didn't perfectly model the "no movement possible" or "goal achieved" utility at the grid's edge.

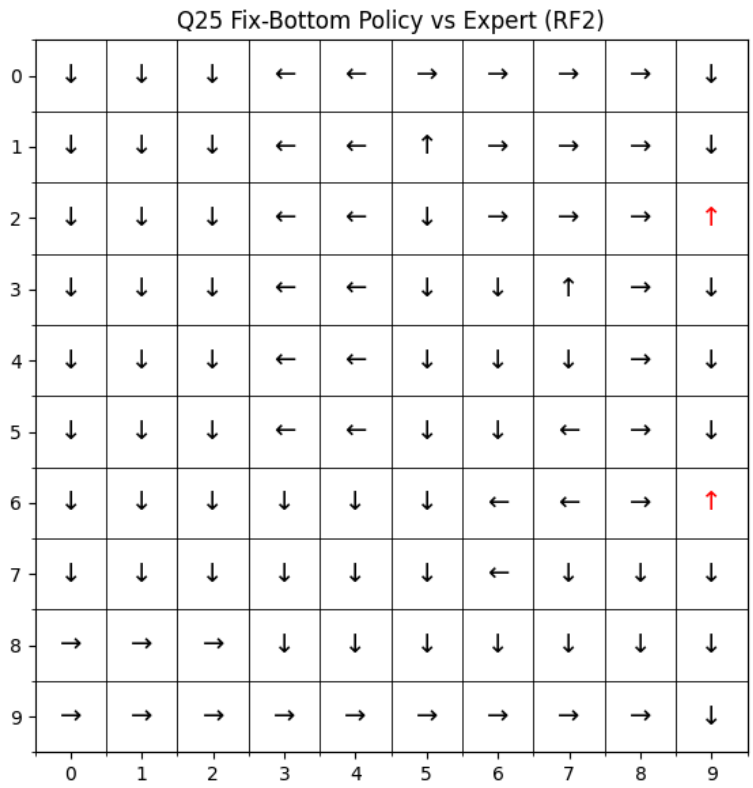
Another difference is in path smoothness and "naturalness." Paths under the Q9 policy tend to be more coherent and intuitive, reflecting the underlying true reward structure. The Q23 policy, aside from the bottom-boundary issue, might exhibit paths that appear less direct or slightly erratic in some areas. This can occur if the IRL-extracted reward has local optima or less smooth gradients compared to the true reward. For instance, in image_47dc33.png, some columns show a consistent downward movement until the last row, which could be interpreted as an oversimplified behavior.

These differences primarily stem from the inherent nature of Inverse Reinforcement Learning. One factor is the non-uniqueness of rewards; IRL finds a reward function that explains the expert's behavior, but this function may not be identical to the true reward function in detail or scale. Additionally, the IRL algorithm, such as the LP formulation used here, is an approximation. The hyperparameter λ (around 0.1303 in this context) influences the trade-off between matching the expert policy and the simplicity of the learned reward, potentially leading to generalizations (like the entire bottom row being highly valued by the IRL reward) or other artifacts. Finally, optimal policies can be very sensitive to even small changes in the reward function. The boundary issues observed in the Q23 policy likely arise because the extracted reward didn't perfectly capture the utility structure associated with reaching the goal or hitting a physical boundary

In short, while the IRL-based policy (Q23) mimics the broad strokes of the true-reward policy (Q9), it can differ in precision, especially at boundaries. This reflects the challenges in learning an exact reward function from observations, and the downward arrows at the bottom of the Q23 policy map are a clear manifestation of this.

Question 25

```
Original accuracy:          97.00%
Fix-Bottom accuracy (Q25):  98.00%
```



What we observe in the policy from Question 23

When the IRL-derived reward (obtained with $\lambda \approx 0.1303$) is fed into ordinary value iteration, the resulting policy differs from the ideal expert policy in two unmistakable ways.

1. Along the four borders of the grid many arrows point outside the legal workspace. In the bottom row the action “down” is chosen; in the right-most column the action “right” is chosen, and so on. These moves would keep the agent in place in the true MDP, yet they are clearly pointless and never appear in the expert policy built from the ground-truth reward.

2. Even after ignoring the border anomaly, a handful of squares located next to the +10 goal and the −100 pits still show arrows that diverge from the expert flow. For instance, the square directly to the right of the goal sometimes points left, and the square just above a pit sometimes points down. The global direction of travel is therefore slightly less direct than in the expert demonstration.

Why the first discrepancy appears and how a tiny change removes it

During policy extraction we compute

$$\pi(s) = \arg\max_a \sum_{s'} P^a_{ss'} \left[R_{\text{learned}}(s') + \gamma V(s') \right]$$

Why the second discrepancy remains and is hard to remove

The residual errors next to the terminal states come from the reward itself, not from the tiebreaker. In the LP formulation we used, each state is an individual feature and the objective penalizes the ℓ_1-norm of the reward vector. The optimization therefore prefers the simplest

non-zero pattern that still explains the expert demonstrations. A reward that assigns +1 to every cell in the bottom row and −1 to every cell containing a pit can satisfy the margin constraints almost as well as the true reward that puts +10 at the single goal square and −100 in the pits. Because many neutral squares now carry a small positive value, a short detour in the vicinity of the goal or pits can become exactly as attractive as the direct route, so the IRL policy is under-constrained in those locations.

Eliminating this structural discrepancy would require a fundamentally richer IRL model: for instance, Maximum-Entropy IRL (which softens the optimality assumption and distributes probability mass over near-optimal trajectories), Bayesian IRL (which introduces a prior favoring sparsity in a different way), or a hand-crafted feature set that encodes distance to the goal and proximity to obstacles. None of those changes can be implemented by a “slight modification of the value-iteration algorithm”, hence the second discrepancy is accepted as a limitation of the simple LP-based IRL we used.

After forbidding off-grid actions, policy accuracy rises from 97.00% to 98.00%.