## 4 Theoretical questions

**Question 8:**

**What is the computational complexity of the VI algorithm, implemented in Question 1?**

The computational complexity of the Value Iteration (VI) algorithm is $O(I \times S \times A)$ where I is the number of iterations, S is the number of states, and A is the number of actions per state.

1. **Number of States (S):**
    - **Explanation:** In Value Iteration, the algorithm must update the value for each state in the Markov Decision Process (MDP). This means that every state in the environment needs to be evaluated in each iteration.Since we have to visit every state at least once per iteration, the algorithm performs S operations per iteration, where S is the total number of states.
2. **Number of Actions (A):**
    - **Explanation:** For each state, the algorithm considers all possible actions the agent can take. This is necessary to determine the maximum expected value from the possible actions, which is part of the Bellman update equation used in value iteration.For each state, evaluating all actions means performing A operations per state, where A is the total number of actions. Thus, the operations for a single iteration across all states is $S \times A$ times.
3. **Number of Iterations (I):**
    - **Explanation:** The algorithm iterates I times to ensure convergence, where I is the number of iterations needed to reach a stable value function.

- **Overall Complexity:** For each iteration, the algorithm performs $S \times A$ times operations to update the value function for all states and actions. Repeating this process for I iterations leads to a total complexity of $O(I \times S \times A)$.

**Question 9:**

**In Question 3, the parameters govern the riskiness of the policy. Explain how each parameter affects the optimal policy.**

In the DiscountGrid layout, we control how the agent makes decisions and handles risk by adjusting three main parameters: the discount factor, noise, and living reward. Here's how each one affects the agent's behavior:

1. **Discount Factor (γ)**:

   **What It Does**: The discount factor tells the agent how much it should care about future rewards versus immediate rewards. A value close to 1 means future rewards are almost as important as immediate ones, while a value close to 0 means the agent focuses more on immediate rewards.

   **Impact on Riskiness**: **High Discount Factor (γ≈1)**: The agent will plan for the long term, often choosing safer paths that might take longer but avoid immediate dangers. This cautious approach is like saving money for a rainy day rather than spending it all now. **Low Discount Factor (γ≈0)**: The agent will focus on what it can gain right away, even if that means taking shortcuts through risky areas. This is like going for quick wins, even if it involves some danger.

2. **Noise**:

   **What It Does**: Noise adds randomness to the agent's actions, meaning sometimes things don’t go exactly as planned. It’s like trying to walk a straight line while being pushed around by the wind.

   **Impact on Riskiness**: **High Noise**: The agent becomes more cautious and avoids risky paths because unexpected things can happen. It sticks to safer routes to minimize bad surprises. **Low Noise**: The agent can reliably predict what will happen, allowing it to take riskier paths since it trusts its actions will lead to expected outcomes.

3. **Living Reward**:

   **What It Does**: This is the reward the agent gets simply for being alive and moving around, regardless of what else it achieves.

**Impact on Riskiness**: **Positive Living Reward**: The agent is motivated to stay alive and avoid risky situations that might end the game. It's like getting paid just for being careful and taking your time. **Negative Living Reward**: The agent feels pressured to find an exit quickly to avoid accumulating penalties, even if that means taking more risks along the way. It's like being fined for every extra minute you spend in a risky area.

**Question 10:**
**In Question 4, can you think about another exploration policy, instead of ε-greedy? How will this policy change the learning process (number of times an action is selected, variability of the estimated Q-values)?**
The Softmax Policy, given by the formula:

$$P(a|s) = \frac{e^{Q(s,a)/\varepsilon}}{\sum_x e^{Q(s,x)/\varepsilon}}$$

Where ϵ controls the level of exploration, when its high value means more exploration.

**Its impact on Learning Process:**

1. **Number of Times an Action is Selected:**

   Actions with higher Q-values have a higher probability of being selected, but there is still some chance for exploration based on the value of ϵ.

   This leads to more informed exploration where actions with higher potential rewards are more frequently tried.

2. **Variability of Estimated Q-values:**

   The variability of Q-values might be lower compared to ε-greedy because the selection is biased towards actions with higher Q-values.

   The agent is more likely to refine its estimates for actions that appear promising, leading to more stable Q-value updates over time.