HW2: 使用價值迭代算法推導最佳政策

1. **價值迭代算法：**
   - 實現**價值迭代算法**來計算最佳政策。
   - 使用該算法推導每個格子的最佳行動。
2. **最佳政策顯示：**
   - 顯示推導出的最佳政策，通過顯示每個狀態（格子）對應的行動來展示最佳政策。
     - 這些行動應該取代之前顯示的隨機行動。
3. **顯示價值函數：**
   - 在執行價值迭代後，更新格子以顯示每個狀態的價值函數 **V(s)**。
   - 每個格子應顯示對應的值，表示在最佳政策下該狀態的期望回報。

其他注意事項：

- 使用 Flask 作為後端，HTML/CSS 作為前端來顯示格子並處理用戶交互。
- 確保 Flask 應用程序是交互式的，用戶可以點擊格子來設置起始點、終點、障礙物，並查看生成的政策。
- 提供可視化的功能，讓用戶能夠清楚地看到隨機政策、價值函數和最佳政策的變化。

prompt:

請幫我用 Python 和 Flask 完成一個互動式網頁程式，實作 Gridworld 環境並使用「價值迭代（Value Iteration）」推導每個狀態的最佳策略與價值函數 V(s)，功能需求如下：

☑ 基本要求

1. 使用者可輸入一個整數 n（範圍 5～9）產生 nxn 網格
2. 使用滑鼠點擊格子依序設定：
   - 起點（綠色）
   - 終點（紅色）
   - 障礙物（灰色，最多 n - 2 個）
   - 點擊已選格子可以取消設定
3. 按下「執行價值迭代」按鈕後執行以下：

🧠 價值迭代邏輯

- 折扣因子 γ = 0.9，步驟獎勵 R = -1
- 終點 V(s) = 0
- 其他格子初始 V(s) = 0
- 障礙格與終點不更新

- 牆與障礙物視為非法行動

🎯 策略推導與路徑追蹤

- 根據每格的最大期望值決定最佳方向（↑ ↓ ←→）
- 從起點開始，依照最佳策略一步步走到終點，記錄「最佳路徑」
- 若中途遇到牆或障礙即中止追蹤
- 把整條最佳路徑用「黃色」背景標示（起點綠、終點紅、障礙灰）

💻 技術規格

- 後端用 Python Flask
- 前端使用 HTML + CSS + JavaScript + jQuery
- 使用 AJAX 傳送資料
- 請回傳完整程式結構（包含 `app.py`、`templates/index.html`、`static/script.js`、`static/styles.css`）

Result:

HW1 + HW2：網格地圖與策略估計

輸入網格大小（5 - 9）： 5 生成網格 隨機策略估值（HW1-2） 執行價值迭代（HW2）

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

策略與價值矩陣

Policy Matrix

↓	↓	↓	↓	
↓	↓	↓	↓	↓
↓	↓	↓	→	↓
→	↓	↓		↓
	→	→	→	

Value Matrix

-5.70	-5.22	-4.69	-4.10	
-5.22	-4.69	-4.10	-3.44	-2.71
-4.69	-4.10	-3.44	-2.71	-1.90
-4.10	-3.44	-2.71		-1.00
	-2.71	-1.90	-1.00	0.00