Unique Paths II

Key idea (DP with obstacles):

Let ways[i][j] be the number of paths to cell (i, j) from (0, 0) moving only right/down and never stepping on a 1.

Transition:

- If obstacleGrid[i][j] == 1, then ways[i][j] = 0 (can't stand here).
- Else, ways[i][j] = ways[i-1][j] + ways[i][j-1] (from top + from left)

Base:

- If the start is blocked (obstacleGrid[0][0] == 1) or the end is blocked, answer is 0.
- Otherwise, ways[0][0] = 1.

Space-optimized (1D DP):

Use a 1D array dp[j] that represents ways for the current row:

- Initialize dp[0] = 1 if the start cell is not an obstacle; otherwise 0.
- For each row i:
  - For each column j from left to right:
    - If obstacleGrid[i][j] == 1:
      set dp[j] = 0 (you cannot land here)
    - Else update dp[j] += dp[j-1] if j > 0 (add ways from the left).
    - (Note: dp[j] already holds the ways from above)

This works because:

- dp[j] before the update = ways from above.
- dp[j-1] = ways from left in the current row.
- Setting to 0 on obstacle kills any paths landing on that cell.