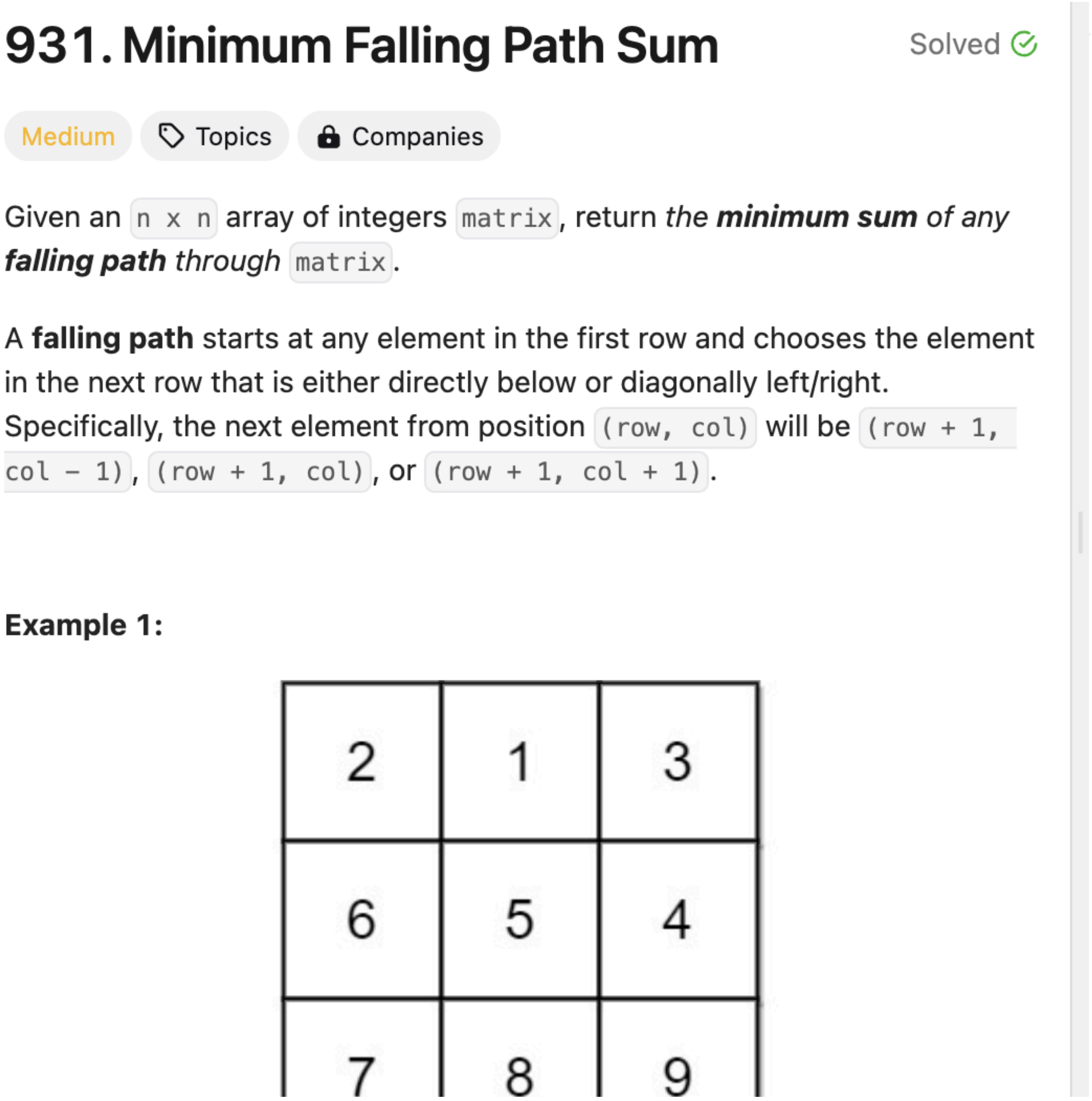

931. Minimum Falling Path Sum

Given an `n x n` array of integers `matrix`, return *the* ***minimum sum*** *of any* ***falling path*** *through* `matrix`.

A **falling path** starts at any element in the first row and chooses the element in the next row that is either directly below or diagonally left/right. Specifically, the next element from position `(row, col)` will be `(row + 1, col - 1)`, `(row + 1, col)`, or `(row + 1, col + 1)`.

Example 1:

2	1	3
6	5	4
7	8	9

```
class Solution {
    func minFallingPathSum(_ matrix: [[Int]]) -> Int {
        let n = matrix.count
        var tp = Array(repeating:0,count: n)
        var dp = Array(repeating:tp,count: n)
        var ans = Int.max

        for i in 0..<n {
            for j in 0..<n {
                if(i == 0) {
                    dp[i][j] = matrix[i][j]
                } else {
                    var t = matrix[i][j]
                    if(j == 0) {
                        t = t + min(dp[i-1][j],dp[i-1][j+1])
                        dp[i][j] = t
                    } else if (j == n-1) {
                        t = t + min(dp[i-1][j],dp[i-1][j-1])
                        dp[i][j] = t
                    } else {
                        t = t +
 min(dp[i-1][j],dp[i-1][j-1],dp[i-1][j+1])
                        dp[i][j] = t
                    }
                }

                if(i == n-1) {
                    ans = min(ans,dp[i][j])
                }
            }
        }
        return ans
    }
 }
```