Unique Paths:

Unique Paths

A robot is located at the top-left corner of a `m x n` grid (marked 'Start' in the diagram below).

The robot can only move either down or right at any point in time. The robot is trying to reach the bottom-right corner of the grid (marked 'Finish' in the diagram below).

How many possible unique paths are there?

Input: m = 3, n = 7
Output: 28

```
def uniquePaths(self, m: int, n: int) -> int:

        dp = [[0]*n for i in range(m)]

        for i in range(m-1,-1,-1):
            for j in range(n-1,-1,-1):
                if i==m-1 and j==n-1:
                    dp[i][j] = 1
                elif i==m-1:
                    dp[i][j] = 1
                elif j==n-1:
                    dp[i][j] = 1
                else:
                    dp[i][j] = dp[i+1][j]+dp[i][j+1]
        return dp[0][0]
```