Autocomplete

public int uniquePaths(int m, int n) {

inpArray[row][0] = 1;

int[][] inpArray = new int[n][m];

for (int row = 0; row < n; row++) {</pre>

for (int col = 0; col < m; col++) {

5

 \odot





LeetCode Explore Problems Mock Contest Articles

□ Discuss (999+)

Submissions

Discuss

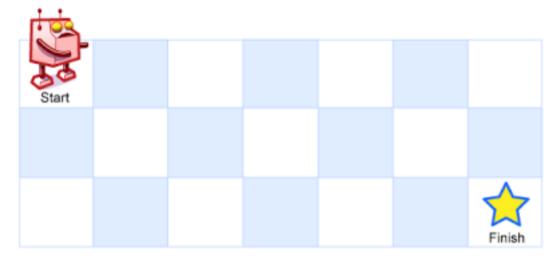
62. Unique Paths

□ 212 □ Add to List □ Share

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?



Above is a 7 x 3 grid. How many possible unique paths are there?

Example 1:

Input: m = 3, n = 2

Output: 3 **Explanation:**

From the top-left corner, there are a total of 3 ways to reach the bottom-right corner:

- 1. Right -> Right -> Down
- 2. Right -> Down -> Right
- 3. Down -> Right -> Right

```
i Java
       class Solution {
  2 🔻
  5 ▼
  6
  8 🔻
  9
 10
 11
 12 ▼
 13 ▼
 14
 15
 16
 17
 18
 19
 20
```

```
inpArray[0][col] = 1;
               for (int row = 1; row < n; row++) {</pre>
                    for (int col = 1; col < m; col++) {
                        inpArray[row][col] = inpArray[row][col - 1] +
      inpArray[row - 1][col];
               return inpArray[n-1][m-1];
Your previous code was restored from your local storage. Reset to default
```