LeetCode Explore Problems Interview New Contest Discuss Store ☆ Premium 6 0

Description Solution Discuss (999+) Submissions

C++ Autocomplete

62. Unique Paths

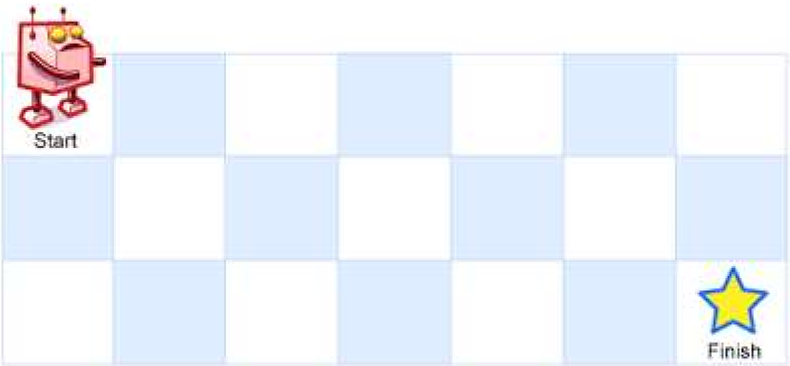
Medium 10656 323 Add to List Share

There is a robot on an `m x n` grid. The robot is initially located at the **top-left corner** (i.e., `grid[0][0]`). The robot tries to move to the **bottom-right corner** (i.e., `grid[m - 1][n - 1]`). The robot can only move either down or right at any point in time.

Given the two integers `m` and `n`, return *the number of possible unique paths that the robot can take to reach the bottom-right corner*.

The test cases are generated so that the answer will be less than or equal to $2 * 10^9$.

Example 1:



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

Example 2:

```
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 -> Down -> Down
2. Down -> Down -> Right
3. Down -> Right -> Down
```

Constraints:

- `1 <= m, n <= 100`

Accepted 1,073,280 Submissions 1,750,768

Seen this question in a real interview before? Yes No

Companies

Related Topics

Similar Questions

Problems Pick One < Prev /1158 Next >

```cpp
class Solution {
public:

    vector<vector<int>> dp;
    int uniquePathsUtil(int m, int n) {
        if(m == 1 || n == 1) return 1;
        if(dp[m][n] != -1)
            return dp[m][n];
        return dp[m][n] = uniquePathsUtil(m-1, n) + uniquePathsUtil(m, n-1);
    }

    int uniquePaths(int m, int n) {
        dp.clear();
        dp.assign(m+1, vector<int>(n+1, -1));
        return uniquePathsUtil(m, n);
    }
};
```

Testcase Run Code Result Debugger

Accepted Runtime: 3 ms

Your input

```
3
7
```

Output

```
28
```

□ Diff

Expected

```
28
```

Example cases Run Code Submit