

Problem 62: Unique Paths

Problem Information

Difficulty: Medium

Acceptance Rate: 66.28%

Paid Only: No

Tags: Math, Dynamic Programming, Combinatorics

Problem Description

There is a robot on an $m \times n$ 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:

$1 \leq m, n \leq 100$

Code Snippets

C++:

```
class Solution {  
public:  
    int uniquePaths(int m, int n) {  
  
    }  
};
```

Java:

```
class Solution {  
    public int uniquePaths(int m, int n) {  
  
    }  
}
```

Python3:

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