

Description

Solution

Discuss (999+)

Submissions

62. Unique Paths

Medium

8046

282

Add to List

Share

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:

Java

Autocomplete

```
1  class Solution {
2
3  private int uniquePaths( int m, int n, HashMap<String, Integer>
    hashMap ) {
4
5      String key = String.valueOf(m) + ',' + n;
6
7      if (m == 1 && n == 1)
8          return 1;
9      else if (m == 0 || n == 0)
10         return 0;
11     else if (hashMap.containsKey(key))
12         return hashMap.get(key);
13     else {
14         hashMap.put(key, uniquePaths(m - 1, n, hashMap) +
            uniquePaths(m, n - 1, hashMap));
15         return hashMap.get(key);
16     }
17 }
18
19 public int uniquePaths(int m, int n) {
20
21     HashMap<String, Integer> hashMap = new HashMap<>();
22
23     return uniquePaths(m, n, hashMap);
24 }
25 }
```

Testcase

Run Code Result

Debugger

Accepted

Runtime: 10 ms

Your input

3

7

Output

28

Diff

Expected

28