Staircase

LeetCode 70 - Climbing Stairs *[easy]*

You are climbing a stair case. It takes **n** steps to reach to the top. Each time you can either climb **1** or **2** steps. In how many *distinct* ways can you climb to the top?

Note:

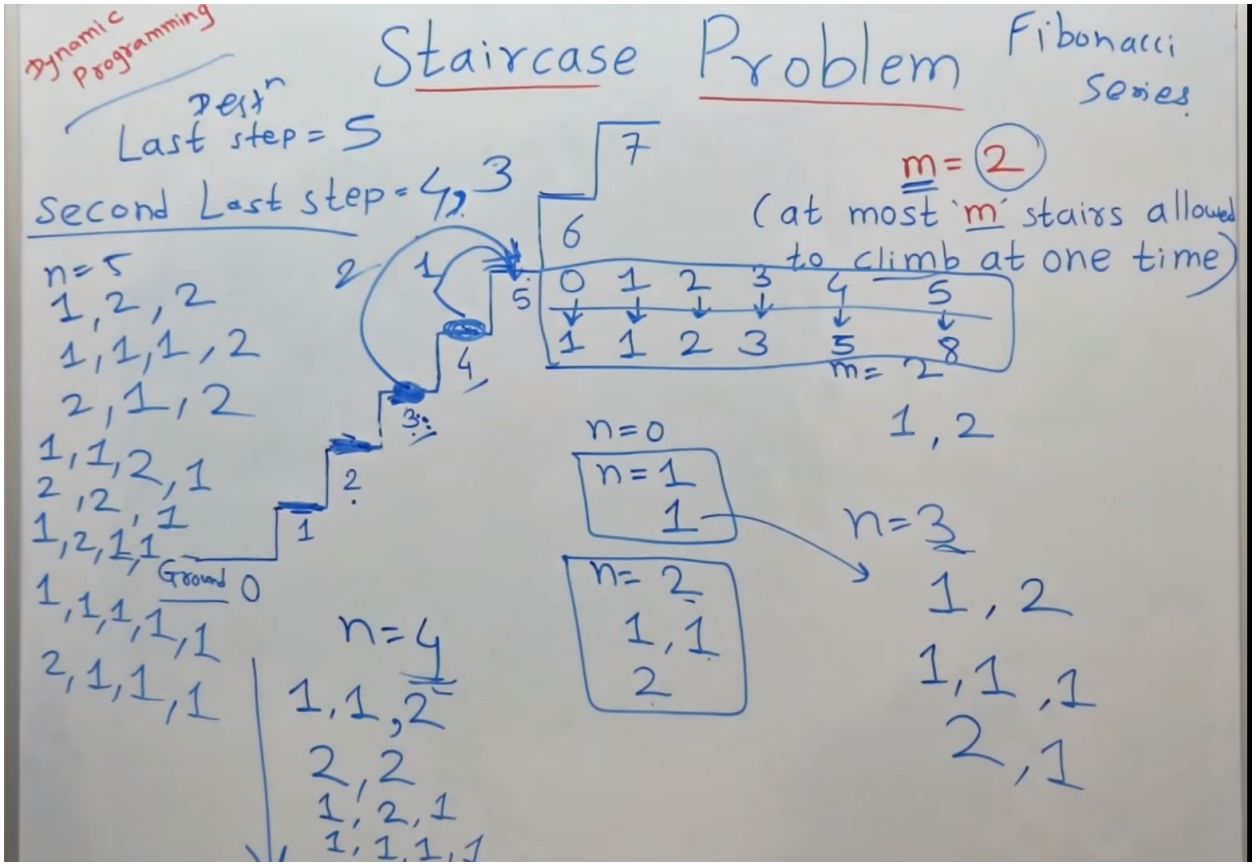
Given n will be a positive integer.

Example 1:

```
Input: 2
Output: 2
Explanation: There are two ways to climb to the top.
--> 1 step + 1 step
--> 2 steps
```

Example 2:

```
Input: 3
Output: 3
Explanation: There are three ways to climb to the top.
--> 1 step + 1 step + 1 step
--> 1 step + 2 steps
--> 2 steps + 1 step
```




```java
class Solution {
    public int climbStairs(int n) {
        int[] dp = new int[n + 1];
        dp[0] = 1;
        dp[1] = 1;
        for(int i = 2; i <= n; i++) {
            dp[i] = dp[i - 1] + dp[i - 2];
        }

        return dp[n];
    }
}
```

[LeetCode 62 - Unique Paths [*medium*]]

[LeetCode 91 - Decode Ways [*medium*]]

[LeetCode 509 - Fibonacci Number [*easy*]]

LeetCode 746 - Min Cost Climbing Stairs [*easy*]

LeetCode 1155 - Number of Dice Rolls With Target Sum [*medium*]