Problem 70: Climbing Stairs

Problem Information

Difficulty: Easy
Acceptance Rate: 53.77%
Paid Only: No
Tags: Math, Dynamic Programming, Memoization

Problem Description

You are climbing a staircase. It takes `n` steps to reach the top.

Each time you can either climb `1` or `2` steps. In how many distinct ways can you climb to the top?

Example 1:

Input: n = 2 **Output:** 2 **Explanation:** There are two ways to climb to the top. 1. 1 step + 1 step 2. 2 steps

Example 2:

Input: n = 3 **Output:** 3 **Explanation:** There are three ways to climb to the top. 1. 1 step + 1 step + 1 step 2. 1 step + 2 steps 3. 2 steps + 1 step

Constraints:

* `1 <= n <= 45`

Code Snippets

C++:

```
class Solution {
public:
```

```
int climbStairs(int n) {

}
};
```

Java:

```
class Solution {
public int climbStairs(int n) {

}
}
```

Python3:

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