70. Climbing Stairs

Easy

179367FavoriteShare

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. 1 step + 1 step  
2. 2 steps

Example 2:

Input: 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

class Solution {

public:

int climbStairs(int n) {

if (n==0) return 0;

int arr[2]={0,1};

int k=1;

while(k<n){

arr[(k+1)%2]=arr[0]+arr[1];

k++;

}

return arr[0]+arr[1];

}

};

Success

[Details](https://leetcode.com/submissions/detail/209640880/)

Runtime: 4 ms, faster than 100.00% of C++ online submissions for Climbing Stairs.

Memory Usage: 8.4 MB, less than 71.48% of C++ online submissions for Climbing Stairs.