070 Climbing Stairs

2018年3月30日 11:42

Question:

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

来自 <https://leetcode.com/problems/climbing-stairs/description/>

你正在爬楼梯。需要 *n* 步你才能到达顶部。

每次你可以爬 1 或 2 个台阶。你有多少种不同的方式可以爬到楼顶呢?

注意：给定 *n* 将是一个正整数。

示例 1：

输入： 2

输出： 2

说明： 有两种方法可以爬到顶端。

1. 1 步 + 1 步

2. 2 步

示例 2：

输入： 3

输出： 3

说明： 有三种方法可以爬到顶端。

1. 1 步 + 1 步 + 1 步

2. 1 步 + 2 步

3. 2 步 + 1 步

Solution for Python3:

Solution for Python3:

3. 2步 + 1步

```python
#Recursive Version
class Solution1:
    def climbStairs(self, n):
        """
        :type n: int
        :rtype: int
        """
        #递归方法n较大时栈太深超时
        return self.climbStairs(n - 1) + self.climbStairs(n - 2)

#Iteration Version 仿照Fibonacci Sequence
class Solution2:
    def climbStairs(self, n):
        """
        :type n: int
        :rtype: int
        """
        l1, l2, r = 1, 1, 1
        for i in range(2, n + 1):
            r = l1 + l2
            l1 = l2
            l2 = r
        return r

#Iteration Version 改进
class Solution3:
    def climbStairs(self, n):
        """
        :type n: int
        :rtype: int
        """
        l1, l2 = 1, 1
        for i in range(2, n + 1):
            l2 += l1
            l1 = l2 - l1
        return l2
```

Solution for C++:

```cpp
class Solution {
public:
    int climbStairs(int n) {
        int l1 = 1, l2 = 1;
        for (int i = 2; i <= n; i++) {
```

```
            l2 += l1;
            l1 = l2 - l1;
        }
        return l2;
    }
};
```