

70. 爬楼梯

题目：70. Climbing Stairs 语言：python3 英文版链接：<https://leetcode.com/problems/climbing-stairs/description/> 中文版链接：<https://leetcode-cn.com/problems/climbing-stairs/description/>

题目分析

2有两种，3有三种，4有五种，5有八种，6有13种，其实这是一个斐波那契数，不过今天我们要用动态规划的思想来求解这个问题。

递归和动态规划都是将原问题拆成多个子问题然后求解，他们之间最本质的区别是，动态规划保存了子问题的解，避免重复计算。

定义一个数组 dp 存储上楼梯的方法数（为了方便讨论，数组下标从 1 开始），dp[i] 表示走到第 i 个楼梯的方法数。

第 i 个楼梯可以从第 i-1 和 i-2 个楼梯再走一步到达，走到第 i 个楼梯的方法数为走到第 i-1 和第 i-2 个楼梯的方法数之和。

$$dp[i] = dp[i - 1] + dp[i - 2]$$

考虑到 dp[i] 只与 dp[i - 1] 和 dp[i - 2] 有关，因此可以只用两个变量来存储 dp[i - 1] 和 dp[i - 2]，使得原来的 O(N) 空间复杂度优化为 O(1) 复杂度。

答案

```
class Solution:
    def climbStairs(self, n: int) -> int:
        if n < 3:
            return n
        # pre1是i-1, pre2是i-2
        pre2, pre1 = 1, 2
        for i in range(2, n):
            pre1, pre2 = pre1 + pre2, pre1
        return pre1
```