

70. 爬楼梯

地址：[爬楼梯](#)

题目：

- English :

70. Climbing Stairs

难度 简单 1733

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`

- 中文：

70. 爬楼梯

难度 简单 1733

假设你正在爬楼梯。需要 n 阶你才能到达楼顶。

每次你可以爬 1 或 2 个台阶。你有多少种不同的方法可以爬到楼顶呢？

注意：给定 n 是一个正整数。

示例 1：

输入： `2`
输出： `2`
解释： 有两种方法可以爬到楼顶。
1. 1 阶 + 1 阶
2. 2 阶

示例 2：

输入： `3`
输出： `3`
解释： 有三种方法可以爬到楼顶。
1. 1 阶 + 1 阶 + 1 阶
2. 1 阶 + 2 阶
3. 2 阶 + 1 阶

思路1：递归改循环

★ 懵逼时候：

★ 是否能暴力解题？

★ 都有什么基本情况？

★ 找最近重复逻辑

★ if else, for, while, recursing

★ 不要用人脑分析 recursing 步骤

★ Do:

★ $n = 1 \rightarrow 1$ 种: 1 阶

★ $n = 2 \rightarrow 2$ 种: 1 阶,

★ $n = 3 \rightarrow 3$ 种: 1 阶, 1 阶 + 2 阶, 2 阶 + 1 阶

★

★ $f(n) = f(n-1) + f(n-2)$: Fibonacci

★ 递归会超时，将递归改为 for 循环。

Code:

```
1  # python3
2
3
4  class Solution:
5      def climbStairs(self, n: int) -> int:
6          # if n == 0:
7          #     return 0
8          # if n == 1:
9          #     return 1
10         # if n == 2:
11         #     return 2
12
13
14         if n <= 2:
15             return n
16
17
18         prepre, pre, cur = 1, 2, 3
19
20
21         for i in range(3, n + 1):
22             cur = pre + prepre
23             prepre = pre
24             pre = cur
25
26
27         return cur
28
29         # 递归代码
30         # return self.climbStairs(n-1) + self.climbStairs(n-2)
```

复杂度分析

- 时间复杂度： $O(n)$
- 空间复杂度： $O(1)$