

## 70. 爬楼梯

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

### 方法一： 动态规划

核心思路：

- 重复子问题:  $f(x)=f(x-1)+f(x-2)$
- 边界条件:  $f(0)=1, f(1)=1$
- 通过转移方程进行动态规划

空间复杂度:  $O(1)$

时间复杂度:  $O(n)$

```
1 class Solution{
2     public:
3         int climbstairs(int n){
4             int p=0, q=1,r=1;
5             for (int i=2;i<=n;++i){
6                 p=q; q=r; r=p+q;
7             }
8             return r;
9         }
10    };
```

### 方法二： 矩阵快速幂

首先我们可以构建这样一个递推关系：

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} f(n) \\ f(n-1) \end{bmatrix} = \begin{bmatrix} f(n) + f(n-1) \\ f(n) \end{bmatrix} = \begin{bmatrix} f(n+1) \\ f(n) \end{bmatrix}$$

因此：

$$\begin{bmatrix} f(n+1) \\ f(n) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n \begin{bmatrix} f(1) \\ f(0) \end{bmatrix}$$

令：

$$M = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

## 如何想到使用矩阵快速幂?

- 如果一个问题可与转化为求解一个矩阵的  $n$  次方的形式, 那么可以用快速幂来加速计算
- 如果一个递归式形如  $f(n) = \sum_{i=1}^m a_i f(n-i)$ , 即齐次线性递推式, 我们就可以把数列的递推关系转化为矩阵的递推关系, 即构造出一个矩阵的  $n$  次方乘以一个列向量得到一个列向量, 这个列向量中包含我们要求的  $f(n)$ 。一般情况下, 形如  $f(n) = \sum_{i=1}^m a_i f(n-i)$  可以构造出这样的  $m \times m$  的矩阵:

$$\begin{bmatrix} a_1 & a_2 & a_3 & \cdots & a_m \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

```
1 class Solution{
2 public:
3 // 通过 struct 构建一个矩阵, 并通过函数定义矩阵运算法则
4     struct Matrics{
5         //[x,y]
6         //[z,t]
7         long long x,y,z,t;
8     };
9     Matrics pow(Matrics, int b){
10         Matrics base =a;
11         Matrics ans={1,0,0,1};
12         while (b!=0){
13             if(b&1!=0)
14                 ans=multiply(ans, base);
15
16                 base=multiply(base, base);
17                 b>>=1;
18         }
19         return ans;
20     }
21
22     Matrics multiply(Matrics a, Matrics b){
23         Matrics c;
24         c.x=a.x*b.x+a.y*b.z;
25         c.y=a.x*b.y+a.y*b.t;
26         c.z=a.z*b.x+a.t*b.z;
27         c.t=a.z*b.y+a.t*b.t;
28         return c;
29     }
30
31     int climStairs(int n){
32         Matrics base={1,1,1,0};
33         Matrics result = pow(base, n);
34         return result.z+result.t;
35     }
```

36 };

- Matrics 结构体中, 需要使用long long, 否则有的数据会超过int能承受的范围。