查阅更多的题解，请点击

Problem

70. Climbing Stairs(Easy)

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
```

Solution

这道题算是比较典型的**动态规划问题**，考虑问题reduce
定义前k个台阶走法个数为DP[k],则显然有

- 第k个台阶是用“One Step”的策略走的，DP[k]=DP[k-1]
- 第k个台阶是用“Two Step”的策略走的，DP[k]=DP[k-2]

所以有DP[k]=DP[k-1]+DP[k-2]

GitHub传送门

```
class Solution {
public:
    int climbStairs(int n) {
        if(n<=2)
            return n;
        vector<int> dp(n+1,1);
        dp[2] = 2;
        for (int i = 3; i <= n;++i){
            dp[i] = dp[i - 1] + dp[i - 2];
        }
        return dp[n];
    }
};
```