

假设你正在爬楼梯。需要 n 步你才能到达楼顶。

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 步 + 1 步
2. 2 步

示例 2：

输入： 3

输出： 3

解释： 有三种方法可以爬到楼顶。

1. 1 步 + 1 步 + 1 步
2. 1 步 + 2 步
3. 2 步 + 1 步

分析，当最后一步为2时，只能走一步，最后一步为1，则可以选择再走一步或者和本步合并为2步走，则可以得到递推式，下一步最后为1等于前面的步数和，为2的等于前面步数为1。假设当前步数中最后一步为1的为 $a(n)$ ，最后一步为2的为 $y(n)$ ，那么下一步中，最后一步为1为

$a(n)+y(n)$ ，最后一步为2的为 $a(n)$ ， $a(n)+y(n)$ 和为 $s(n)$ ， $a(n+1)+y(n+1)$ 为 $s(n+1)$ ，则 $s(n+1)=s(n)+a(n)$ ，因为为2的等于前面步数为1，所以 $a(n)=s(n-1)$ ，则递推式出来 $s(n+1)=s(n)+s(n-1)$ 。当数据为1，2时返回1，2，大于则按递推式运算。

```
class Solution {
public:
    int climbStairs(int n) {
        if(n<3){
            return n;
        }
        int a[2]={1,2};
        for(int i=2;i<n;++i){
            int h=a[1];
            a[1]=a[0]+a[1];
            a[0]=h;
        }
        return a[1];
    }
};
```