

爬楼梯

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

解题思路：

爬楼梯问题即为一个斐波那契数列问题

可以使用递归模板求解，但存在过大数值导致超出时间限制问题

通过采用备忘录的方式，记录已经做完的计算，用这种方法来避免重复计算。
这其实也是一种动态规划。

```
class Solution {
public:
    /*
        int climbStairs(int n) {
            if(n==1||n==2)
                return n;
            return climbStairs(n-1)+climbStairs(n-2);
        }
    */
};
```

```
    }  
    */  
    int fun(int n,int* arr)  
    {  
        if(arr[n]!=-1)//已经存在在记录中的直接返回  
        {  
            return arr[n];  
        }  
        int res=fun(n-1,arr)+fun(n-2,arr);//不存在的通过递归计算  
        arr[n]=res;  
        return res;  
    }  
    int climbStairs(int n)  
    {  
        int arr[100];  
        memset(arr,-1,sizeof(int)*100);  
        arr[1]=1;  
        arr[2]=2;  
        return fun(n,arr);  
    }  
};
```