

面试题：跳台阶

2018/8/10 8:55:57

题目描述

一只青蛙一次可以跳上1级台阶，也可以跳上2级。求该青蛙跳上一个n级的台阶总共有多少种跳法（先后次序不同算不同的结果）。

解题思路

可以观察到，从第3个数开始，每个数的值都等于前连个数之和。

同时，定义 $f(0)=0$, $f(1)=1$, $f(2)=2$;

则 $f(3)=f(2)+f(1)=2$;

... 依次类推,

$$f(n)=f(n-1)+f(n-2)$$

该问题实质是斐波那契数列求和，递推公式为 $f(n)=f(n-1)+f(n-2)$;

可以考虑，小青蛙每一步跳跃只有两种选择：一是再跳一级阶梯到达第 i 级阶梯，此时小青蛙处于第 $i-1$ 级阶梯；或者再跳两级阶梯到达第 i 级阶梯，此时小青蛙处于第 $i-2$ 级阶梯。

于是， i 级阶梯的跳法总和依赖于前 $i-1$ 级阶梯的跳法总数 $f(i-1)$ 和前 $i-2$ 级阶梯的跳法总数 $f(i-2)$ 。因为只有两种可能性，所以， $f(i)=f(i-1)+f(i-2)$;

依次类推，可以递归求出 n 级阶梯跳法之和。

C++ (迭代)

递归慢，迭代效率更高

```
1 class Solution {
2 public:
3     int jumpFloor(int number) {
4         if(number==1||number==2){
5             return number;
6         }
7         int jumpFib=0;
8         int NumberMinusOne=2;
9         int NumberMinusTwo=1;
10        for(int i=3;i<=number;i++){
11            jumpFib =
NumberMinusOne+NumberMinusTwo;
12            NumberMinusTwo = NumberMinusOne;
13            NumberMinusOne = jumpFib;    //迭代
```

```
14         }
15         return jumpFib;
16     }
17 };
```

Java (递归)

```
1 public class Solution {
2     public int JumpFloor(int target) {
3         int result = 0;
4         if(target == 0){
5             result = 0;
6         }else if(target == 1){
7             result = 1;
8         }else if(target == 2){
9             result = 2;
10        }else{
11            result = JumpFloor(target - 1) +
JumpFloor(target - 2);
12        }
13        return result;
14    }
15 }
```

Java (迭代)

```
1 public class Solution {
```

```
2     public int JumpFloor(int target) {
3         if(target < 0)
4             return 0;
5         int[] fib = {0,1,2};
6         if(target < 3)
7             return fib[target];
8         int total = 0;
9         int FirstElem = 1;
10        int SecondElem = 2;
11        for(int i = 3; i <= target; i++){
12            total = FirstElem + SecondElem;
13            FirstElem = SecondElem;
14            SecondElem = total; //迭代
15        }
16        return total;
17    }
18 }
```