

剑指10 斐波那契数列/青蛙跳台阶问题

我：抬手就是递归，然后存在重复计算，直接超时。

递归法：

原理：把 $f(n)$ 问题的计算拆分成 $f(n-1)$ 和 $f(n-2)$ 两个子问题的计算，并递归，以 $f(0)$ 和 $f(1)$ 为终止条件。

缺点：大量重复的递归计算，例如 $f(n)$ 和 $f(n-1)$ 两者向下递归需要各自计算 $f(n-2)$ 的值。他妈的 $O(2^n)$

优化：记忆化递归——新建数组存储函数值，避免重复计算。

动态规划法：看官方题解下评论区。// 递归本质上是基于分治思想的从顶至底的解法。借助记忆化递归思想，可应用动态规划从底至顶求取

快速幂法：TODO看官方题解下评论区。

算法学习笔记(4)：快速幂 - Pecco的文章 - 知乎

<https://zhuanlan.zhihu.com/p/95902286>

```
1 // 标准答案
2
3 f(n) , 代码如下所示。
4 class Solution {
5 public:
6     int fib(int n) {
7         int MOD = 1000000007;
8         if (n < 2) {
9             return n;
10        }
11        int p = 0, q = 0, r = 1;
12        for (int i = 2; i <= n; ++i) {
13            p = q;
14            q = r;
15            r = (p + q)%MOD;
16        }
17        return r;
18    }
19 };
20
```

1 2021-10-18 微软面试，要求递归且不重复

```
2
3 class Solution {
4 public:
5
```

```

6  int fib(int n) {
7      if(n==0) return 0;
8      vector<int>dp(101,-1);
9      dp[1]=1;
10     dp[2]=1;
11     reverse(n,dp);
12     return dp[n];
13 }
14 int reverse(int n,vector<int>& dp){
15     if(n==1||n==2) return 1;
16
17     if(dp[n]==-1)
18         dp[n]=reverse(n-1,dp)+reverse(n-2,dp);
19     dp[n]=dp[n]%1000000007;
20     return dp[n];
21 }
22 };
23
24
25
26

```