

第19题目——斐波那契数列

写一个函数，输入 n ，求斐波那契（Fibonacci）数列的第 n 项（即 $F(N)$ ）。斐波那契数列的定义如下：

$F(0) = 0$, $F(1) = 1$ $F(N) = F(N - 1) + F(N - 2)$, 其中 $N > 1$. 斐波那契数列由 0 和 1 开始，之后的斐波那契数就是由之前的两数相加而得出。

答案需要取模 $1e9+7$ (1000000007)，如计算初始结果为：1000000008，请返回 1。

示例 1:

输入: $n = 2$

输出: 1

示例 2:

输入: $n = 5$

输出: 5

提示:

$0 \leq n \leq 100$

动态规划（1）

首先，我们的初始状态，分为三种情况：

$f(0)=0$, $f(1)=1$

方程式：

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

终止状态：

$f(n)$

```
int fib(int n){

    if(n==0)
    {
        return 0;
    }
    int x=1,y=1,sum=0;
    //斐波那契数列：从第三项开始，每项的值=前面两项的和
    for(int i=3;i<=n;i++)//从第三项开始计算
    {
        sum=(x+y)%1000000007;//取前面两项的和
        x=y;                //x后移一位
        y=sum;              //y后移一位
    }
    return y;
}
```

代码解析：

初始值 $x=1,y=1$;

若输入的 $n=0$;返回0;

若输入的是1，不满足循环条件，返回y;

输入的是2，不满足循环条件，返回y;

输入值 ≥ 3 时，执行循环体，模拟数据如下：

sum	2	3	5	8	13	21	...
x	1	2	3	5	8	13	...
y	2	3	5	8	13	21	...

每一次都使 $y=x+y$ （相当于在斐波那契数列中后移一位）；然后使 $x=y$ （也后移一位）。

循环结束后返回该数列的最后一项即可，因此返回y。

递归法（1）

```
int arr[101] = {0};

int fib(int n){
    if (0 == n || n == 1){
        return n;
    }
}
```

```
if (0 != arr[n]){  
    return arr[n]; //返回已储存的值  
}  
arr[n] = (fib(n - 1) + fib(n - 2)) % 1000000007; //将得到的项存入数组  
return arr[n];  
}
```

没什么好说的，递归，言简意赅。