

## 剪绳子-2 (测试数据过大 原有方法可能造成时间溢出)

给你一根长度为  $n$  的绳子，请把绳子剪成整数长度的  $m$  段 ( $m, n$  都是整数， $n > 1$  并且  $m > 1$ )，每段绳子的长度记为  $k[0], k[1] \dots k[m-1]$ 。请问  $k[0] * k[1] * \dots * k[m-1]$  可能的最大乘积是多少？例如，当绳子的长度是 8 时，我们把它剪成长度分别为 2、3、3 的三段，此时得到的最大乘积是 18。

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

```
/*class Solution {
public:
    int cuttingRope(int n) {
        if(n<=3)
            return n-1;
        //n>1
        //n=2 分割为 1+1 1*1=1;
        //n=3 分割为 1+2 1*2=2;
        //if(n==4) return 4;
        //n>5 部分尽量分出 3 使得乘积最大
        int cout=n/3;
        int res=0;
        if(n%3==0)
        {
            res=pow(3,cout);
            //return res % 1000000007; //可以直接分割为 3 的整数倍
        }
        else if(n%3==1)
        {
            cout--;
            res=(pow(3,cout)*4);
            //return res % 1000000007; //最后一组与多余的 1 合并为 4
        }
        else
        {
            res=(pow(3,cout)*2);
            //return res % 1000000007; //剩余 2 个组合一组
        }
        return res % 1000000007;
    }
};*/
class Solution {
public:
    int cuttingRope(int n) {
        if(n <= 3) return n - 1;
        if(n == 4) return 4;
```

```
long res = 1;
while(n > 4)
{
    res *= 3;
    res %= 1000000007;
    n -= 3;
}
// 最后 n 的值只有可能是：2、3、4。而 2、3、4 能得到的最大乘积恰恰就是自身
// 因为 2、3 不需要再剪了（剪了反而变小）；4 剪成 2x2 是最大的，2x2 恰巧等于
4
return res * n % 1000000007;
}
};
```