

第35题——剪绳子II

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。

示例 1:

输入：2

输出：1

解释：2 = 1 + 1, 1 × 1 = 1

示例 2:

输入：10

输出：36

解释：10 = 3 + 3 + 4, 3 × 3 × 4 = 36

提示:

$2 \leq n \leq 1000$

解题思路:

这道题和剪绳子1的题干一样,但是这道题的难点在于,数值太大了,我们存不下这样的数值,所以要取巧:

评论区大佬解法(1)

首先,如果绳子长度 <4 ,我们返回 $n-1$ 即可

然后,进入一个while循环.不断计算乘积result

每次计算前 $n-=3$;因为我们会 $\times 3$,所以要先计算 $n=n-3$;

循环的结束条件是: $n \leq 4$

也就是说当绳子只剩下4或更短时,我们没必要再去剪断它,因为乘积不会大于绳子长度了

所以我们直接返回剩余绳子长度和result的乘积就好了

```
class Solution {
public:
    int cuttingRope(int n) {
        if(n<4)
            return n-1;
        long result = 1;
        while(n > 4){
            n -= 3;
            result *= 3;
            result %= 1000000007;
        }
        return (int)(n * result % 1000000007);
    }
};
```