# [剑指 Offer 14- II. 剪绳子 II]--动态规划--大整数相乘处理

class Solution {  
 private:  
 /\*  
 //循环取余法解决大整数相乘问题  
 int MulRecursive(int a, int b){  
 int min\_num=min(a,b);  
 int max\_num=max(a,b);  
 int res=0;  
 int remain=min\_num;//剩余倍数  
 int factor=max\_num;//被乘数  
 while(remain){  
 if(remain&1){  
 //奇数  
 factor%=1000000007;  
 res+=factor;  
 res%=1000000007;  
 remain-=1;  
 }else{  
 //偶数  
 factor<<=1;//扩大一倍  
 remain>>=1;//缩小一倍  
 }  
   
 }  
 return res;  
 }  
 \*/  
 //快速求幂 O(logn)复杂度--优化  
 long long int mod;  
 int MulRecursive(long long int base, long long int num){  
 long long res = 1;  
 while(num > 0){  
 if((num & 1) == 1){//num为奇数时 //位与0001  
 res \*= base;  
 res %= mod;  
 num-=1;  
 }  
 //base \*= base;//num为偶数时,base平方  
 base\*=base;  
 base %= mod;  
 num >>= 1;//经典的除2取整,多的一在if中表示给res了  
 }  
 return (int)res;  
 }  
  
  
  
public:  
 int cuttingRope(int n) {  
 //初始化mod  
 mod=1000000007;  
 //特殊情况处理  
 if(n==2)return 1;  
 else if(n==3)return 2;  
 //动态规划  
 vector<int>dp(n+1,0);  
 //初始化  
 for(int i=1;i<=3;i++)  
 dp[i]=i;  
  
 //遍历  
 for(int i=4;i<=n;i++){  
 int temp=0;  
 for(int j=1;j<=i/2;j++){  
 //temp=max(temp,dp[j]\*dp[i-j]);  
 //循环取余法  
 temp=max(temp,MulRecursive(dp[j],dp[i-j]));  
 }  
 dp[i]=temp%1000000007;  
 }  
  
 return dp[n]%1000000007;  
 }  
};

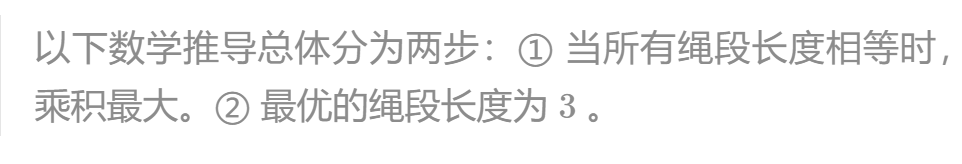
**大整数乘法，还是有问题的**

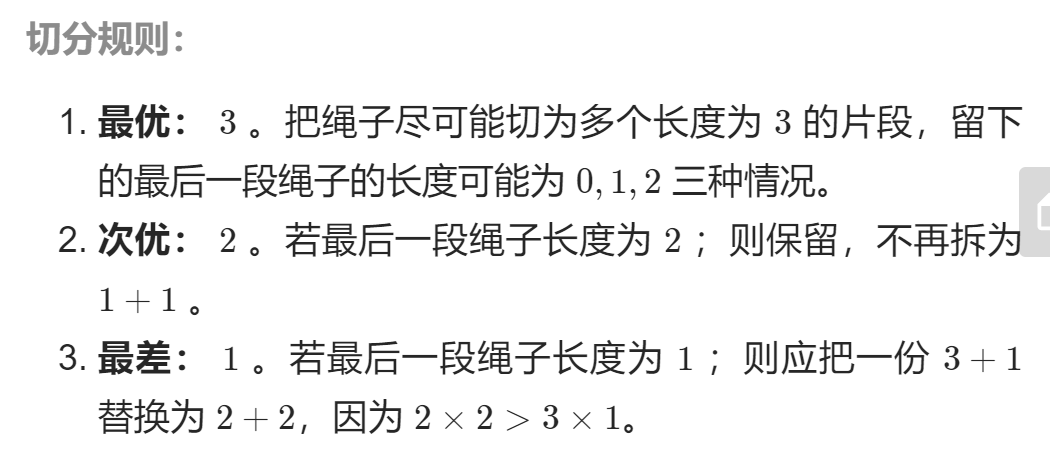
## ？？这道题如果用动态规划的话，整数越界的问题几乎灭有办法德得到解决。

## 编程中的数学问题：

### 大佬的详细推导：

<https://leetcode-cn.com/problems/jian-sheng-zi-ii-lcof/solution/mian-shi-ti-14-ii-jian-sheng-zi-iitan-xin-er-fen-f/>





## 该题的收获：大整数相乘处理--循环取余||快速幂乘法

### 快速求幂：

//快速求幂 O(logn)复杂度  
private: long pow(long base, int num){  
 long res = 1;  
 while(num > 0){  
 if((num & 1) == 1){//num为奇数时 //位与0001  
 res \*= base;  
 res %= mod;  
 }  
 base \*= base;//num为偶数时,base平方  
 base %= mod;  
 num >>= 1;//经典的除2取整,多的一在if中表示给res了  
 }  
 return res;  
 }  
};  
  
作者：pilgrim-r  
链接：https://leetcode-cn.com/problems/jian-sheng-zi-ii-lcof/solution/liang-chong-tan-lan-suan-fa-by-pilgrim-r-3fbg/  
来源：力扣（LeetCode）  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

### 循环取余法

int MulRecursive(int a, int b){  
 int min\_num=min(a,b);  
 int max\_num=max(a,b);  
 int res=0;  
 int remain=min\_num;//剩余倍数  
 int factor=max\_num;//被乘数  
 while(remain){  
 if(remain&1){  
 //奇数  
 factor%=1000000007;  
 res+=factor;  
 res%=1000000007;  
 remain-=1;  
 }else{  
 //偶数  
 factor<<=1;//扩大一倍  
 remain>>=1;//缩小一倍  
 }  
   
 }  
 return res;  
 }

## 如何算法实现加减乘除运算：

<https://writings.sh/post/algorithm-basic-math-computations#%E5%BF%AB%E9%80%9F%E5%B9%82>