

343. 整数拆分

给定一个正整数 n ，将其拆分为至少两个正整数的和，并使这些整数的乘积最大化。返回你可以获得的最大乘积。

示例 1: 输入: 2

输出: 1

解释: $2 = 1 + 1, 1 \times 1 = 1$ 。

示例 2:

输入: 10

输出: 36

解释: $10 = 3 + 3 + 4, 3 \times 3 \times 4 = 36$ 。

说明: 你可以假设 n 不小于 2 且不大于 58。

本题目的变数在于将其拆分为至少两个正整数的和**可能为两个整数也可以为3个整数等**，**没有明确的限制**这个就是需要分析的地方

1、确定**dp数组(dp table)**以及**下标的含义**
dp[i]:分拆数字i,可以得到的最大乘积为dp[i]

2、确定递推公式

可以想dp[i]最大乘积是怎么得到的呢?

其实可以从1遍历j, 然后有两种渠道得到dp[i]

1、 $j*(i-j)$ 2、 $j*dp[i-j]$ 取最大

为什么j不拆分呢?

因为j是从1开始遍历的，拆分j的情况，在遍历j的过程中其实都计算过了

递推公式: $dp[i] = \max(dp[i], \max(i*(i-j), dp[i-j]*j));$

3、dp初始化

$dp[2] = 1$ ($dp[0], dp[1]$ ，拆分0,1的最大乘积是多少这是无解的)

4、遍历顺序的确定

由于dp[i]的确定来自于dp[i-j],所以是从前往后遍历

由于dp[2]已经求出来了所以**从dp[3]开始求出因此i从3枚举，j从1开始枚举正好dp[3]可以从dp[2]数值求出来**

```
1 for(int i=3;i<=n;i++){
2     for(int j=1;j<i-1;j++){
3         dp[i]=max(dp[i],max(dp[i-j]*j,(i-j)*j));
4     }
5 }
```

```
1 class Solution {
2 public:
3     int integerBreak(int n) {
4         vector<int> dp(n+1);
5         dp[2]=1;
6         for(int i=3;i<=n;i++){
7             for(int j=1;j<=i-1;j++){
8                 dp[i]=max(dp[i],max((i-j)*j,dp[i-j]*j));
9             }
10        }
11        return dp[n];
12    }
13 };
```