**请你在理解“整数拆分问题”“最大连续子序列和问题”的基础上，说明它们的“重叠子问题”性质和最优子结构性质。**

**整数拆分问题**

f(n, k)表示正整数n拆分成最大数为k的拆分方案数

首先分析一下边界情况，当n == 1或k == 1的时候，很明显只有1种拆分方案

若k == n， 比如f(5, 5)，那么其中必然有一个拆分方案是n = n， 其余的拆分方案个数就是f(n, k - 1)。所以当n == k的时候，等于f(n, k - 1) + 1。

再如果k > n， 那么拆分方案个数是和f(n, n)相等的。

最后分析k < n这种最常规的情况，这里有个思路，对于一个数n，它的拆分方案个数应该是等于**拆出一个数k的方案个数 + 没有拆出数k的方案个数**，这里的这个思路有点像01背包。

子问题为求解拆出一个数k的方案个数和没有拆出数k的方案个数。

**最大连续子序列和**

不需要按照PPT的思路维护DP数组，浪费空间。

PPT并没有使用最优子结构。

PPT代码中并没有使用dp[0]---dp[j-1]之间的值，我们只需要维护单点maxSum和nowSum即可。

即子问题为求解nowSum的最大值

If(nowSum + a[j] > 0)

nowSum = nowSum + a[j];

Else

nowSum = 0;

maxSum = max(maxSum , nowSum);

重复n次。

此为最优子结构。

其实不难看出，这里并非典型的DP。在状态转移时我们多出了一个限制条件，即序列和必须大于0，小于0的序列和无意义。在此基础上我们就不需要维护每一点的最大值，也就是dp数组，我们不需要考虑将原有结果加回来这个过程。

也可以理解为：对于每一个子问题，我们都采用同样的决策，那么就不需要考虑这些子问题具体是什么了，也就是不需要存储dp数组。

第十三周 动态规划-算法过程

1. 说明求解“最长公共子序列问题”的状态转移方程的设计思路。
2. 填充用于保存各级子问题解的动态规划数组。参考 PPT59页

序列为 X： ATTCGAG

Y: CTCACAGG

3.给出最长公共子序列的长度，以及具体组成。

1. 状态转移方程设计思路：

①最优子结构

设 X=(x1,x2,.....xn) 和 Y={y1,y2,.....ym} 是两个序列，将 X 和 Y 的最长公共子序列记为LCS(X,Y)

找出LCS(X,Y)是最优化问题。

1）如果 xn=ym，即X的最后一个元素与Y的最后一个元素相同，此时该元素一定位于公共子序列中。只需要找：LCS(Xn-1，Ym-1)

LCS(Xn-1，Ym-1)是原问题的子问题。

2）如果xn != ym，此时有两个子问题：LCS(Xn-1，Ym) 和 LCS(Xn，Ym-1)

LCS(Xn-1，Ym)表示：最长公共序列可以在(x1,x2,....x(n-1)) 和 (y1,y2,...yn)中找。

LCS(Xn，Ym-1)表示：最长公共序列可以在(x1,x2,....xn) 和 (y1,y2,...y(n-1))中找。

求解上面两个子问题，得到的公共子序列中最长的，就是 LCS（X,Y）。

②重叠子问题

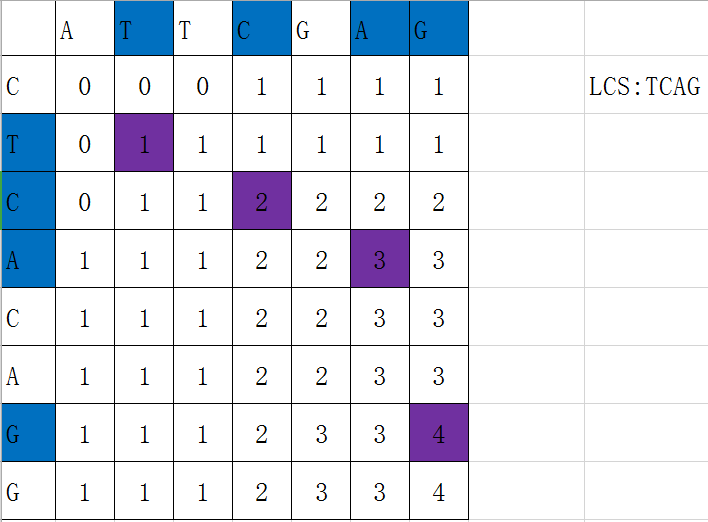
子问题有 ❶LCS(Xn-1，Ym-1)    ❷LCS(Xn-1，Ym)    ❸LCS(Xn，Ym-1)

重叠部分举例：

第二个子问题：LCS(Xn-1，Ym) 包含了问题❶LCS(Xn-1，Ym-1)

故可得转移方程为：LCS=max{LCS(Xn-1，Ym)，LCS(Xn，Ym-1)}

1. DP表格：



3.LSC长度为4 TCAG，已在上图标出。