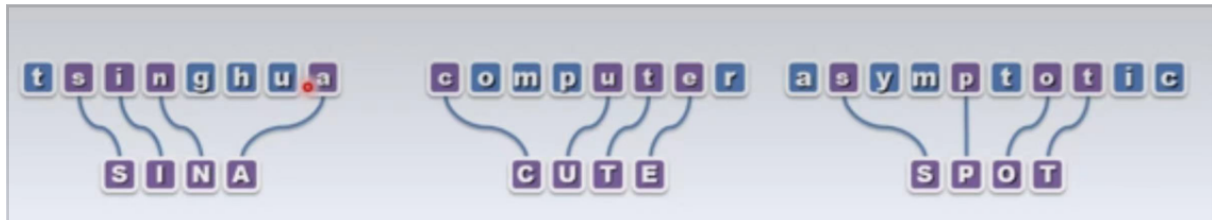


01-F-6 最长公共子序列 LCS & 递归LCS

#数据结构邓神

LCS 递归



子序列：在序列中若干字符（可以不连续），按照原来的相对次序，拼成新的序列

最长公共子序列(Longest Common Subsequence)（相对两个序列而言）：两个序列的公共子序列中的长者

可能有多个，可能有歧义

❖ 最长公共子序列 (Longest Common Subsequence)：两个序列公共子序列中的最长者

可能有多个

可能有歧义



其中 dana 和 dana 都是长度为 4 的子序列

递归 LCS 算法

Q: 对于序列 $A[0,n]$ 和 序列 $B[0,m]$, $LCS(A,B)$ 无非三种情况

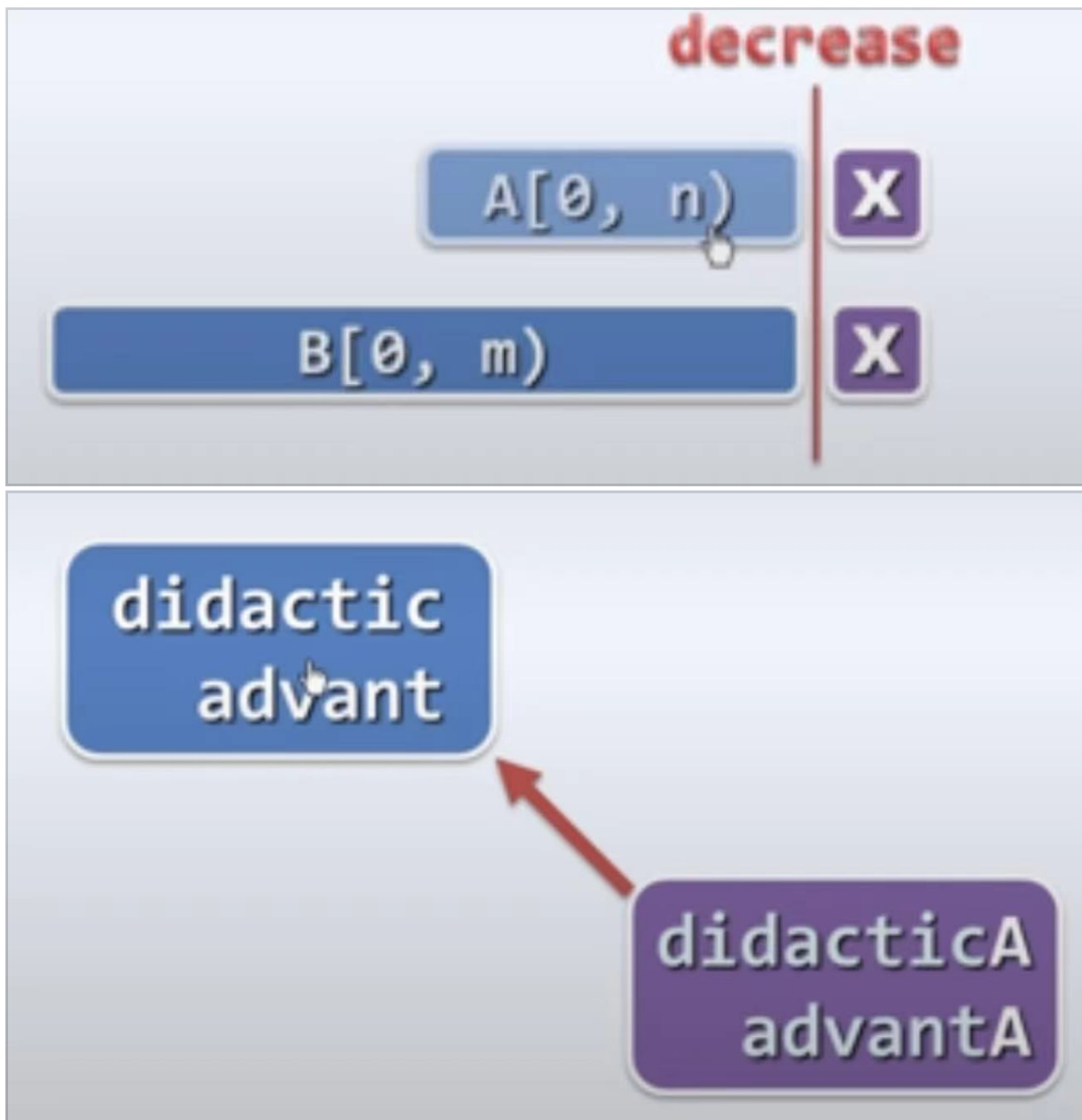
0: 如果 $n = -1$ 或者 $m = -1$ 则取空序列 "" // 递归基

1: 若 $A[n] = 'X' = B[m]$, 则取 $LCS(A[0,n], B[0,m]) + 'X'$ // 减而治之

2: $A[n] \neq B[m]$, 则在 $LCS(A[0,n], B[0,m])$ 与 $LCS(A[0,n], B[0,m])$ 中取长者

把问题切分为一个平凡的子问题和一个规模减少1的规模缩减的问题

1: 关键就是找到一个相同元素后回头去找



2. 如果不想等，可能是 Y 对这个公共序列没有贡献，那就不要了，也有可能是 X，我们两个都尝试一下

2) $A[n] \neq B[m]$, 则在 $LCS(A[0, n], B[0, m])$ 与 $LCS(A[0, n), B[0, m])$ 中取更长者

//分而治之

