最长公共子序列

子序列

设序列 X, Z,

$$X = \langle x_1, x_2, \ldots, x_m \rangle$$

$$Z = \langle z_1, z_2, \ldots, z_k \rangle$$

若存在 X 的元素构成的严格递增序列 $\langle x_{i_1}, x_{i_2}, \ldots, x_{i_k} \rangle$ 使得

$$z_j = x_{i_j}, j = 1,2,\ldots,k$$

则称 Z 是 X 的子序列

X 与 Y 的公共子序列 Z：Z 是 X 和 Y 的子序列

子序列的长度：子序列的元素个数

最长公共子序列

问题：给定序列

$$X = < x_1, x_2, \dots, x_m >$$

$$Y = < y_1, y_2, \dots, y_n >$$

求 X 和 Y 的最长公共子序列

实例：

X: A B C B D A B

Y: B D C A B A

最长公共子序列：B C B A, 长度4

蛮力算法

不妨设 $m \le n$，$|X| = m$，$|Y| = n$

算法：依次检查 X 的每个子序列在 Y 中是否出现

时间复杂度：

每个子序列 $O(n)$ 时间

X 有 2^m 个子序列

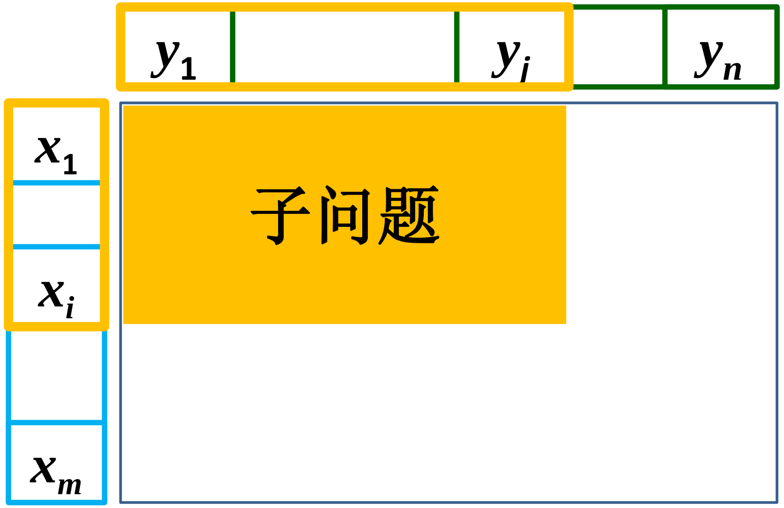
最坏情况下时间复杂度：$O(n\,2^m)$

子问题界定

参数 i 和 j 界定子问题

X 的终止位置是 i，Y 的终止位置是 j

$X_i=<x_1, x_2, \ldots, x_i>$，$Y_j=<y_1, y_2, \ldots, y_j>$



子问题间的依赖关系

设 $X=<x_1,x_2,\ldots,x_m>$, $Y=<y_1,y_2,\ldots,y_n>$, $Z=<z_1,z_2,\ldots,z_k>$ 为X和Y的LCS，那么

(1) 若 $x_m=y_n \Rightarrow z_k=x_m=y_n$, 且

Z_{k-1} 是X_{m-1}与Y_{n-1} 的LCS；

(2) 若 $x_m\neq y_n$， $z_k\neq x_m \Rightarrow$

Z 是 X_{m-1} 与 Y 的 LCS；

(3) 若 $x_m\neq y_n$, $z_k\neq y_n \Rightarrow$

Z 是 X 与 Y_{n-1} 的 LCS.

满足优化原则和子问题重叠性

优化函数的递推方程

令 X 与 Y 的子序列

$X_i = <x_1, x_2, \ldots, x_i>$， $Y_j = <y_1, y_2, \ldots, y_j>$

$C[i,j]$： X_i 与 Y_j 的 LCS 的长度

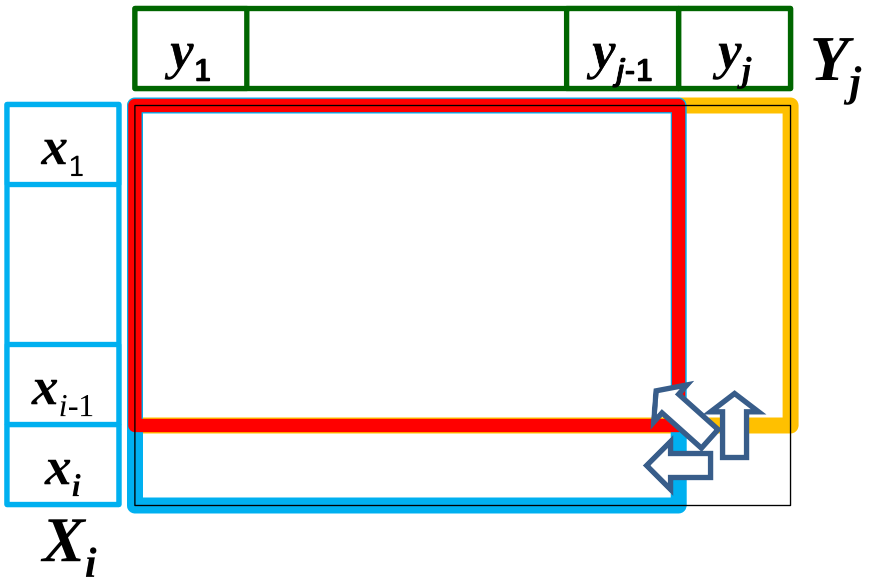
$$C[i,j] = \begin{cases} 0 & 若\ i=0 或\ j=0 \\ C[i-1,j-1]+1 & 若\ i,j>0, x_i = y_j \\ \max\{C[i,j-1], C[i-1,j]\} & 若\ i,j>0, x_i \neq y_j \end{cases}$$

标记函数

标记函数：$B[i, j]$，值为↖、←、↑

$C[i, j]=C[i-1,j-1]+1$: ↖ $C[i, j]=C[i,j-1]$: ←

$C[i, j]=C[i-1,j]$: ↑



伪码

算法 LCS (X, Y, m, n)

```
for i←1 to m do  C[i,0] ← 0
for i←1 to n do  C[0,i] ← 0
for i←1 to m do
    for j←1 to n do
        if X[i]=Y[j]
        then C[i, j] ←C[i-1,j-1]+1
             B[i, j] ←"↖"
        else if C[i-1, j] ≥ C[i,j-1]
        then C[i, j]←C[i-1,j]
             B[i, j]←"↑"
        else C[i, j]←C[i,j-1]
             B[i, j]←"←"
```

初值

子问题 X_i, Y_j

追踪解

算法 Structure Sequence(B, i, j)

输入：$B[i, j]$

输出：X与Y的最长公共子序列

1. if i=0 or j=0 then return //序列为空
2. if $B[i, j]$ = "↖"
3. then 输出$X[i]$
4. Structure Sequence(B, $i-1$, $j-1$)
5. else if $B[i,j]$= "↑"
6. then Structure Sequence (B, $i-1$, j)
7. else Structure Sequence (B, i, $j-1$)

标记函数的实例

输入：X = < A, B, C, B, D, A, B >,

Y = < B, D, C, A, B, A >,

	1	2	3	4	5	6
1	B[1,1]=↑	B[1,2]=↑	B[1,3]=↑	B[1,4]=↖	B[1,5]=←	B[1,6]=↖
2	B[2,1]=↖	B[2,2]=←	B[2,3]=←	B[2,4]=↑	B[2,5]=↖	B[2,6]=←
3	B[3,1]=↑	B[3,2]=↑	B[3,3]=↖	B[3,4]=←	B[3,5]=↑	B[3,6]=↑
4	B[4,1]=↑	B[4,2]=↑	B[4,3]=↑	B[4,4]=↑	B[4,5]=↖	B[4,6]=←
5	B[5,1]=↑	B[5,2]=↑	B[5,3]=↑	B[5,4]=↑	B[5,5]=↑	B[5,6]=↑
6	B[6,1]=↑	B[6,2]=↑	B[6,3]=↑	B[6,4]=↖	B[6,5]=↑	B[6,6]=↖
7	B[7,1]=↑	B[7,2]=↑	B[7,3]=↑	B[7,4]=↑	B[7,5]=↑	B[7,6]=↑

解：X[2],X[3], X[4], X[6], 即 B, C, B, A

算法的时空复杂度

计算优化函数和标记函数：

赋初值，为$O(m)+O(n)$

计算优化、标记函数迭代 $\Theta(mn)$次，

循环体内常数次运算，时间为$\Theta(mn)$

构造解：

每步缩小 X 或Y的长度，时间$\Theta(m+n)$

算法时间复杂度： $\Theta(mn)$

空间复杂度：$\Theta(mn)$

小结

- 最长公共子序列问题的建模
- 子问题边界的界定
- 递推方程及初值，优化原则判定
- 伪码
- 标记函数与解的追踪
- 时间复杂度