# Plus longue sous-séquence commune

Référence : T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein
*Introduction to algorithms*

2011-2012

Étant donnés deux mots $X = x_1 \cdots x_m$ et $Y = y_1 \cdots y_n$ sur un alphabet $\Sigma$, on cherche à savoir quel est le plus long sous-mot commun à $X$ et $Y$.

La solution naïve est d'énumérer tous les sous-mots de $X$, et de regarder ensuite s'ils sont aussi sous-mots de $Y$, mais cette solution est inefficace : il y a $2^m$ sous-mots de $X$.

Notation – Si $U$ est un mot de longueur $p$, on note $U_{p-j}$ le préfixe de $U$ de longueur $p - j$.

Un algorithme plus efficace pourra être déduit du théorème de structure suivant :

**Théorème 1 : de structure des plus longues sous-séquences communes**
*Supposons que $Z = z_1 \cdots z_k$ est une plus longue sous-séquence commune à $X$ et $Y$. Alors :*

*(i) Si $x_m = y_n$, alors $z_k = x_m (= y_n)$ et $Z_{k-1}$ est une plus longue sous-séquence commune à $X_{m-1}$ et $Y_{n-1}$.*

*(ii) Si $x_m \neq y_n$ alors*

$$(z_k \neq x_m) \Rightarrow Z \text{ est une plus longue sous-séquence commune à } X_{m-1} \text{ et } Y.$$

*(iii) Si $x_m \neq y_n$ alors*

$$(z_k \neq y_n) \Rightarrow Z \text{ est une plus longue sous-séquence commune à } X \text{ et } Y_{n-1}.$$

*Démonstration.* C'est plutôt facile :

(i) $x_m = y_n$. Si $z_k \neq x_m$, alors $Zx_m$ est un sous-mot commun à $X$ et $Y$, ce qui contredit l'hypothèse de maximalité. D'où $z_k = x_m = y_n$.
De plus, $Z_{k-1}$ est un sous-mot commun à $X_{n-1}$ et $Y_{m-1}$. Supposons qu'il existe un sous-mot commun $W$ de longueur plus grande que $k - 1$.
Alors, $Wx_m$ est de longueur plus grande que $k$, et $Wx_m$ est un sous-mot commun à $X$ et $Y$, d'où une contradiction.
Donc $Z_{k-1}$ est une plus longue sous-séquence commune à $X_{m-1}$ et $Y_{n-1}$.

(ii) $x_m \neq y_n$. Supposons $z_k \neq x_m$. Alors $Z$ est un sous-mot de $X_{m-1}$, et donc *a fortiori un sous-mot commun à $X_{m-1}$ et $Y$.*
*Supposons qu'il existe un sous-mot $W$ commun à $X_{m-1}$ et $Y$, de longueur plus grande. Alors $W$ est* a fortiori un sous-mot commun à $X$ et $Y$, ce qui contredit l'hypothèse de maximalité de $Z$.

(iii) *cf* (ii)

□

Connaissant la plus longue sous-séquence commune au mot vide et n'importe quel mot, on peut en déduire une formule de récurrence sur la longueur de la plus longue sous-séquence commune de $X$ et $Y$. Si on note $c[i,j]$ la longueur de la plus longue sous-séquence commune à $X_i$ et $Y_j$, on a :

$$c[i,j] = \begin{cases} 0 & \text{si } i = 0 \text{ ou } j = 0 \\ c[i-1,j-1]+1 & \text{si } i,j > 0 \text{ et } x_i = y_j \\ \max(c[i,j-1], c[i-1,j]) & \text{si } i,j > 0 \text{ et } x_i \neq y_j \end{cases} \quad (1)$$

Cette formule pourrait nous donner un algorithme récursif pour calculer $c[m,n]$, mais sa complexité serait à nouveau exponentielle. On va plutôt utiliser un algorithme de *programmation dynamique*.

Cet algorithme va calculer les uns après les autres les valeurs des $c[i,j]$, en utilisant la formule de récurrence, en remplissant le tableau par lignes.

**Algorithm 1** Longueur-PLSC

**Préconditions:** $X, Y$ mots sur $\Sigma$
  $m \leftarrow$ Longueur$(X)$
  $n \leftarrow$ Longueur$(Y)$
  **pour** $i = 1$ to $m$ **faire**
    $c[i,0] \leftarrow 0$
  **ruop**
  **pour** $j = 0$ to $n$ **faire**
    $c[0,j] \leftarrow 0$
  **ruop**
  **pour** $i = 1$ to $m$ **faire**
    **pour** $j = 1$ to $n$ **faire**
      **si** $x_i = y_j$ **alors**
        $c[i,j] \leftarrow c[i-1,j-1]+1$
        $b[i,j] \leftarrow$ " $\nwarrow$ "
      **sinon si** $c[i-1,j] \geqslant c[i,j-1]$ **alors**
        $c[i,j] \leftarrow c[i-1,j]$
        $b[i,j] \leftarrow$ " $\uparrow$ "
      **sinon**
        $c[i,j] \leftarrow c[i,j-1]$
        $b[i,j] \leftarrow$ " $\leftarrow$ "
      **is**
    **ruop**
  **ruop**

  **retourner** $c$ et $b$

La matrice $b$ sert à noter "d'où on vient" pour calculer la valeur $c[i,j]$ d'une case. Grâce à ce tableau, on peut donc retrouver la plus longue sous-séquence commune à $X$ et $Y$ par l'algorithme :

**Algorithm 2** PLSC

**Préconditions:** $X$ mot sur $\Sigma$, $b$ matrice calculée par Longueur-PLSC, $i, j$ indices.

**si** $i = 0$ ou $j = 0$ **alors**

    **afficher** ()

**is**

**si** $b[i, j] =$ " $\nwarrow$ " **alors**

    PLSC$(b, X, i - 1, j - 1)$

    **afficher** $x_i$

**sinon si** $b[i, j] =$ " $\uparrow$ " **alors**

    PLSC$(b, X, i - 1, j)$

**sinon**

    PLSC$(b, X, i, j - 1)$

**is**

**Théorème 2**
*La procédure Longueur-PLSC a une complexité en $\mathcal{O}(mn)$, et la procédure PLSC a une complexité en $\mathcal{O}(m+n)$.*

*Démonstration.* Dans Longueur-PLSC, on a deux boucles "pour" imbriquées, et dans ces boucles des opérations en $\mathcal{O}(1)$. D'où la complexité.

Dans PLSC, on réalise un "chemin direct" dans notre matrice $(i + 1) \times (j + 1)$, et donc ce chemin ne peut avoir plus de $i + j$ étapes. □

Exemple – On cherche la plus longue sous-séquence commune aux chaînes `BDCABA` et `ABCBDAB`. Le tableau construit par Longueur-PLSC est :

| | $j$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|---|
| $i$ | | $y_j$ | $B$ | $D$ | $C$ | $A$ | $B$ | $A$ |
| 0 | $x_i$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | $A$ | 0 | 0 $\uparrow$ | 0 $\uparrow$ | 0 $\uparrow$ | 1 $\nwarrow$ | 1 $\leftarrow$ | 1 $\nwarrow$ |
| 2 | $B$ | 0 | 1 $\nwarrow$ | 1 $\leftarrow$ | 1 $\leftarrow$ | 1 $\uparrow$ | 2 $\nwarrow$ | 2 $\leftarrow$ |
| 3 | $C$ | 0 | 1 $\uparrow$ | 1 $\uparrow$ | 2 $\nwarrow$ | 2 $\leftarrow$ | 2 $\uparrow$ | 2 $\uparrow$ |
| 4 | $B$ | 0 | 1 $\nwarrow$ | 1 $\uparrow$ | 2 $\uparrow$ | 2 $\uparrow$ | 3 $\nwarrow$ | 3 $\leftarrow$ |
| 5 | $D$ | 0 | 1 $\uparrow$ | 2 $\nwarrow$ | 2 $\uparrow$ | 2 $\uparrow$ | 3 $\uparrow$ | 3 $\uparrow$ |
| 6 | $A$ | 0 | 1 $\uparrow$ | 2 $\uparrow$ | 2 $\uparrow$ | 3 $\nwarrow$ | 3 $\uparrow$ | 4 $\nwarrow$ |
| 7 | $B$ | 0 | 1 $\nwarrow$ | 2 $\uparrow$ | 2 $\uparrow$ | 3 $\uparrow$ | 4 $\nwarrow$ | 4 $\uparrow$ |

On suit les flèches (chemin grisé) et on note les lettres correspondant aux flèches obliques : `BCBA`.