# Développement : plus longue sous-séquence commune

Pierron Théo

10 avril 2014

**Définition** On dit que $Z$ est une sous-séquence de $X$ ssi on peut obtenir $Z$ en effaçant des lettres de $X$, ie il existe des indices $(i_j)_j$ strictement croissants tels que $z_j = x_{i_j}$.

Étant données deux chaînes de caractères $X$ et $Y$, on cherche une plus longue sous-séquence qui est commune à $X$ et $Y$. (on note $X_i$ le $i$-ème préfixe de $X$, de même pour $Y$ et $Z$).

Théorème Structure des solutions optimales *Si $X = x_1 \dots x_m$, $Y = y_1 \dots y_n$ et $Z = z_1 \dots z_k$ est une PLSC de $X$ et $Y$ alors :*

- *Si $x_m = y_n$ alors $z_k = x_m$ et $Z_{k-1}$ est une PLSC de $X_{m-1}$ et $Y_{n-1}$*
- *Si $x_m \neq y_n$ et $z_k \neq x_m$ alors $Z$ est une PLSC de $X_{m-1}$ et $Y$*
- *Si $x_m \neq y_n$ et $z_k \neq y_n$ alors $Z$ est une PLSC de $X$ et $Y_{n-1}$*

*Démonstration.*

- Si $x_m = y_n$, supposons que $z_k \neq x_m$. Alors $Zx_m$ est une sous-séquence de $X$ et $Y$ strictement plus longue que $Z$, ce qui est absurde. Donc $z_k = x_m$.
  Si $Z_{k-1}$ n'est pas une PLSC de $X_{m-1}$ et $Y_{n-1}$, soit $W$ une telle PLSC. $Wz_k$ est une sous-séquence de $X$ et $Y$ et
  $$|Wz_k| = 1 + |W| > 1 + |Z_{k-1}| = |Z_k|$$
  Absurde. Donc $Z_{k-1}$ est une PLSC de $X_{m-1}$ et $Y_{n-1}$.
- Si $x_m \neq y_n$ et $z_k \neq x_m$, $Z$ est une sous-séquence de $X_{m-1}$ et $Y$. Supposons que ce n'est pas une PLSC, ie il existe $W$ sous-séquence de $X_{m-1}$ et $Y$ telle que $|W| > |Z|$.
  $W$ est aussi une sous-séquence de $X$ et $Y$, strictement plus grande que $Z$, ce qui est absurde. Donc $Z$ est une PLSC de $X_{m-1}$ et $Y$.
- Si $x_m \neq y_n$ et $z_k \neq y_n$, on est dans le cas symétrique. ∎

On note $c[i, j]$ la longueur d'une PLSC de $X_i$ et $Y_j$. On a alors la formule de récurrence :

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

Cette relation conduit naturellement à l'algorithme 1, qui utilise la programmation dynamique.

La complexité est en $O(mn)$ en espace et en temps.

On peut alors reconstruire une PLSC à partir de $b$, via l'algorithme 2, de complexité $O(m+n)$ au pire.

**Algorithme 1:** CoûtPLSC$(X, Y)$

**Entrées** : Deux mots $X$ et $Y$
**Sorties** : Le tableau $c$, et un tableau $b$ permettant de reconstruire une PLSC de $X$ et $Y$

**pour** $i = 0 \dots m$ **faire**
  $c[i, 0] \leftarrow 0$
**pour** $j = 0 \dots n$ **faire**
  $c[0, j] \leftarrow 0$
**pour** $i = 1 \dots m$ **faire**
  **pour** $j = 1 \dots n$ **faire**
    **si** $x_i = y_j$ **alors**
      $c[i, j] \leftarrow 1 + c[i-1, j-1]$ $b[i, j] \leftarrow' \nwarrow'$
    **sinon**
      **si** $c[i-1, j] > c[i, j-1]$ **alors**
        $c[i, j] \leftarrow c[i-1, j]$ $b[i, j] \leftarrow' \uparrow'$
      **sinon**
        $c[i, j] \leftarrow c[i, j-1]$ $b[i, j] \leftarrow' \leftarrow'$
**retourner** $(c, b)$

**Algorithme 2:** PLSC$(b, X, i, j)$

**Entrées** : Le tableau $b$, la chaîne $X$ et deux indices $i, j$
**Sorties** : L'affichage d'une PLSC associée à $b$

**si** $ij = 0$ **alors**
  **retourner** *Fin*
**sinon**
  **si** $b[i, j] =' \nwarrow'$ **alors**
    PLSC$(b, X, i-1, j-1)$ Affiche $x_i$
  **sinon**
    **si** $b[i, j] =' \uparrow'$ **alors**
      PLSC$(b, X, i-1, j)$
    **sinon**
      PLSC$(b, X, i, j-1)$