

Plus long sous-mot commun

Thierry Lecroq
Université de Rouen

On considère

- un mot x de longueur m ;
- un mot y de longueur n .

On veut déterminer

- la longueur (unique) des plus longs sous-mots commun à x et y ;
- un ou tous les plus longs sous-mots commun à x et y .

On définit

- $smc(x,y)$ = longueur d'un plus long sous-mot commun à x et y
- $Smc(x,y)$ = ensemble des plus longs sous-mots commun à x et y
- $d_{smot}(x,y) = |x| + |y| - 2 \times smc(x,y)$

Cela revient à considérer la distance avec les coûts suivants (pour $a, b \in A$ et $a \neq b$) :

- $Sub(a, a) = 0$
- $Dél(a) = Ins(a) = 1$
- $Sub(a, b) > Dél(a) + Ins(b) = 2$

Calcul par programmation dynamique

On considère une table à 2 dimensions S ($m+1$ lignes et $n+1$ colonnes) :

$$S[i,j] = \begin{cases} 0 & \text{si } i = -1 \text{ ou } j = -1 \\ smc(x[0..i],y[0..j]) & \text{sinon} \end{cases}$$

$$S[m-1,n-1] = smc(x,y)$$

Calcul par programmation dynamique

On considère une table à 2 dimensions S ($m+1$ lignes et $n+1$ colonnes) :

$$S[i,j] = \begin{cases} S[i-1,j-1] + 1 & \text{si } x[i] = y[j] \\ \max \{ S[i-1,j], S[i,j-1] \} & \text{sinon} \end{cases}$$

$$S[m-1,n-1] = \text{smc}(x,y)$$

Exemple

<i>S</i>	<i>j</i>	-1	0	1	2	3	4	5	6	7	8	9
<i>i</i>		<i>y</i> [<i>j</i>]	C	A	G	A	T	C	A	G	A	G
-1	<i>x</i> [<i>i</i>]											
0	A											
1	G											
2	C											
3	T											
4	G											
5	A											

Exemple

<i>S</i>	<i>j</i>	-1	0	1	2	3	4	5	6	7	8	9
<i>i</i>		<i>y</i> [<i>j</i>]	C	A	G	A	T	C	A	G	A	G
-1	<i>x</i> [<i>i</i>]	0	0	0	0	0	0	0	0	0	0	0
0	A	0	0	1	1	1	1	1	1	1	1	1
1	G	0	0	1	2	2	2	2	2	2	2	2
2	C	0	1	1	2	2	2	3	3	3	3	3
3	T	0	1	1	2	2	3	3	3	3	3	3
4	G	0	1	1	2	2	3	3	3	4	4	4
5	A	0	1	2	2	3	3	3	4	4	5	5

Exemple

<i>S</i>	<i>j</i>	-1	0	1	2	3	4	5	6	7	8	9
<i>i</i>		<i>y</i> [<i>j</i>]	C	A	G	A	T	C	A	G	A	G
-1	<i>x</i> [<i>i</i>]	0	0	0	0	0	0	0	0	0	0	0
0	A	0	0	1	1	1	1	1	1	1	1	1
1	G	0	0	1	2	2	2	2	2	2	2	2
2	C	0	1	1	2	2	2	3	3	3	3	3
3	T	0	1	1	2	2	3	3	3	3	3	3
4	G	0	1	1	2	2	3	3	3	4	4	4
5	A	0	1	2	2	3	3	3	4	4	5	5

Exemple

S	j	-1	0	1	2	3	4	5	6	7	8	9
i		$y[j]$	C	A	G	A	T	C	A	G	A	G
-1	$x[i]$	0	0	0	0	0	0	0	0	0	0	0
0	A	0	0	1	1	1	1	1	1	1	1	1
1	G	0	0	1	2	2	2	2	2	2	2	2
2	C	0	1	1	2	2	2	3	3	3	3	3
3	T	0	1	1	2	2	3	3	3	3	3	3
4	G	0	1	1	2	2	3	3	3	4	4	4
5	A	0	1	2	2	3	3	3	4	4	5	5

$x = \text{AGCTGA}$

$y = \text{CAGATCAGAG}$

2 plus longs sous-mots communs :

AGCGA

AGTGA

algo SMC-SIMPLE(x, m, y, n)

pour $i \leftarrow -1$ à $m-1$ **faire**

$S[i, -1] \leftarrow 0$

pour $j \leftarrow 0$ à $n-1$ **faire**

$S[-1, j] \leftarrow 0$

pour $i \leftarrow 0$ à $m-1$ **faire**

si $x[i] = y[j]$ **alors**

$S[i, j] \leftarrow S[i-1, j-1] + 1$

sinon

$S[i, j] \leftarrow \max \{S[i-1, j], S[i, j-1]\}$

retourner $S[m-1, n-1]$

algo UN-SMC(x, m, y, n)

$z \leftarrow \varepsilon$

$(i, j) \leftarrow (m-1, n-1)$

tantque $i \neq -1$ et $j \neq -1$ faire

si $x[i] = y[j]$ alors

$z \leftarrow x[i] \cdot z$

sinon si $S[i-1, j] > S[i, j-1]$ alors

$i \leftarrow i - 1$

sinon

$j \leftarrow j - 1$

retourner z

Complexité

- temps et espace : $O(mn)$
- espace $O(\min\{m,n\})$ pour le calcul de $smc(x,y)$

algo SMC-COLONNE(x, m, y, n)

pour $i \leftarrow -1$ à $m-1$ **faire**

$C_1[i] \leftarrow 0$

pour $j \leftarrow 0$ à $n-1$ **faire**

$C_2[-1] \leftarrow 0$

pour $i \leftarrow 0$ à $m-1$ **faire**

si $x[i] = y[j]$ **alors**

$C_2[i] \leftarrow C_1[i-1] + 1$

sinon

$C_2[i] \leftarrow \max \{C_2[i-1], C_1[i]\}$

$C_1 \leftarrow C_2$

retourner C_1

Calcul d'un plus long sous-mot commun en espace linéaire

- On détermine un sommet du graphe d'édition de la forme $(k-1, \lfloor n/2 \rfloor - 1)$ (avec $0 \leq k \leq m$) par lequel passe un chemin optimal.
- Ensuite on calcule les 2 portions du chemin :
 - de $(-1, -1)$ à $(k-1, \lfloor n/2 \rfloor - 1)$;
 - de $(k-1, \lfloor n/2 \rfloor - 1)$ à $(m-1, n-1)$.

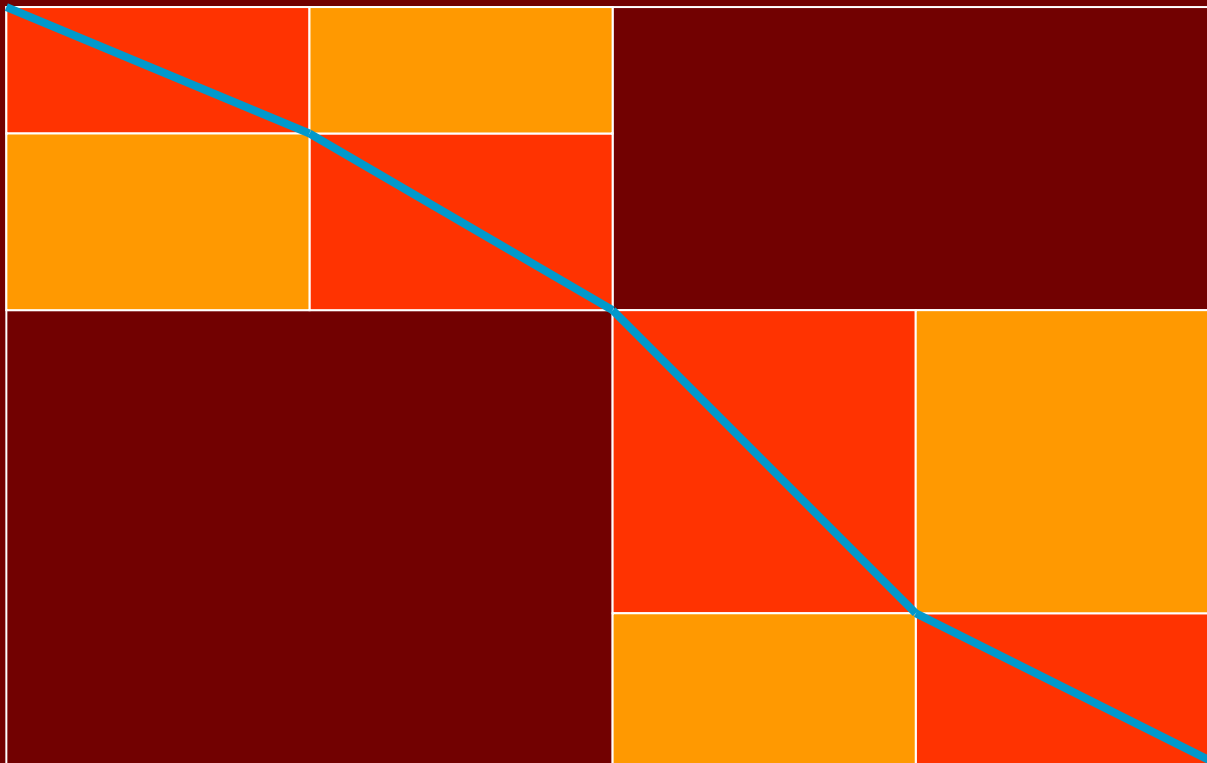
Cela revient à calculer :

- un plus long sous-mot commun u à $x[0..k-1]$ et $y[0.. \lfloor n/2 \rfloor - 1]$
- un plus long sous-mot commun v à $x[k..m-1]$ et $y[\lfloor n/2 \rfloor .. n-1]$
- retourner uv

$(-1,-1)$

$\lfloor n/2 \rfloor$

k



$(m-1,n-1)$

k est tel que la quantité

$$smc(x[0..k-1], y[0..\lfloor n/2 \rfloor - 1]) + \\ smc(x[k..m-1], y[\lfloor n/2 \rfloor .. n-1])$$

est maximale.

Pour calculer k :

- on commence par calculer la colonne d'indice $\lfloor n/2 \rfloor - 1$ avec $\text{SMC-COLONNE}(x, m, y, \lfloor n/2 \rfloor)$;
- pour la deuxième partie des calculs on mémorise en plus des pointeurs vers la colonne du milieu.

$(-1, -1)$

$\lfloor n/2 \rfloor$

k

$(m-1, n-1)$

```

algo SMC( $x,m,y,n$ )
  si  $m = 1$  et  $x[0] \in \text{alph}(y)$  alors
    retourner  $x[0]$ 
  sinon si  $n = 1$  et  $y[0] \in \text{alph}(x)$  alors
    retourner  $y[0]$ 
  sinon si  $m \in \{0,1\}$  ou  $n \in \{0,1\}$  alors
    retourner  $\varepsilon$ 
   $C_1 \leftarrow \text{SMC-COLONNE}(x,m,y,\lfloor n/2 \rfloor)$ 
  pour  $i \leftarrow -1$  à  $m-1$  faire
     $P_1[i] \leftarrow i+1$ 
  pour  $j \leftarrow \lfloor n/2 \rfloor$  à  $n-1$  faire
     $(C_2[-1], P_2[-1]) \leftarrow (0,0)$ 
    pour  $i \leftarrow 0$  à  $m-1$  faire
      si  $x[i] = y[j]$  alors
         $(C_2[i], P_2[i]) \leftarrow (C_1[i-1] + 1, P_1[i-1])$ 
      sinon si  $C_1[i] > C_2[i-1]$  alors
         $(C_2[i], P_2[i]) \leftarrow (C_1[i], P_1[i])$ 
      sinon
         $(C_2[i], P_2[i]) \leftarrow (C_2[i-1], P_2[i-1])$ 
     $(C_1, P_1) \leftarrow (C_2, P_2)$ 
   $k \leftarrow P_1[m-1]$ 
   $u \leftarrow \text{SMC}(x[0..k-1], k, y[0..\lfloor n/2 \rfloor - 1], \lfloor n/2 \rfloor)$ 
   $v \leftarrow \text{SMC}(x[k..m-1], m-k, y[\lfloor n/2 \rfloor..n-1], n-\lfloor n/2 \rfloor)$ 
  retourner  $uv$ 

```

Complexité

temps : $\Theta(mn)$

espace : $\Theta(m)$ ($\sum_i (mn)/2^i \leq 2mn$)

Référence

D. S. Hirschberg,

A linear space algorithm for computing maximal
common subsequences,

Communications of the ACM, 18(6):341-343,
1975.