

I Plus grande sous-chaîne commune

Si w et s sont deux chaînes de caractères, on dit que w est une sous-chaîne (ou : sous-séquence) de s si les lettres de w apparaissent dans l'ordre dans s (mais pas forcément consécutivement, contrairement à un facteur). Par exemple, "hiats" est une sous-chaîne de "this is a test", mais pas un facteur.

Soit s et t deux chaînes de caractères. Écrire un algorithme efficace permettant de trouver la longueur d'une plus grande sous-chaîne commune à s et t . Par exemple, si $s = \text{"this is a test"}$ et $t = \text{"another try"}$, il faut renvoyer 4 (correspondant à "thet" qui est la plus grande sous-chaîne de s et t).

II Palindromes

Une chaîne de caractères est un palindrome si elle se lit de la même manière de gauche à droite et de droite à gauche. Par exemple, "kayak" est un palindrome.

1. Écrire une fonction en C pour déterminer si une chaîne de caractères est un palindrome.
2. Par programmation dynamique, écrire une fonction `longest` telle que `longest s` renvoie la longueur d'une plus longue sous-chaîne de s qui est un palindrome. Par exemple, `longest "kbcaydak"` doit renvoyer 5 ("kayak" est bien une sous-chaîne de "kbcaydak" de taille 5).
Quelle est sa complexité ?
3. Faire de même, par programmation dynamique, pour le plus long facteur (avec des lettres consécutives).
Remarque : même s'il n'est pas demandé ici, il existe un algorithme (de Manacher) en complexité linéaire pour ce problème.

III Fenêtre minimum

Soient s et t deux chaînes de caractères. Écrire un algorithme efficace donnant un facteur de s de longueur minimum contenant toutes les lettres de t .

Par exemple, si $s = \text{"adobecodebanc"}$ et $t = \text{"abc"}$ alors l'algorithme doit renvoyer "banc".

Indice : on pourra utiliser deux « pointeurs » (indices de s).