ISM MP, Informatique
Année 2024/2025

TP Informatique 11

Exercice 1

On définit la *distance de Levenshtein* ou *distance d'édition* notée lev entre deux chaînes de caractères X et Y par

$$\mathrm{lev}(\mathrm{X},\mathrm{Y}) = \begin{cases} \max(|\mathrm{X}|,|\mathrm{Y}|) & \text{si } \min(|\mathrm{X}|,|\mathrm{Y}|)=0 \\ \mathrm{lev}(\mathrm{X}[1:],\mathrm{Y}[1:]) & \text{si } \mathrm{X}[0]=\mathrm{Y}[0] \\ 1+\min\left(\mathrm{lev}(\mathrm{X}[1:],\mathrm{Y}),\mathrm{lev}(\mathrm{X},\mathrm{Y}[1:]),\mathrm{lev}(\mathrm{X}[1:],\mathrm{Y}[1:])\right) & \text{sinon} \end{cases}$$

On peut montrer que la distance de Levenshtein est le nombre minimal d'opérations élémentaires (substitution, insertion ou suppression d'un caractère) pour transformer une chaîne de caractères en une autre.

Pour deux chaînes de caractères X, Y de tailles respectives n et m, on pose

$$\forall (i,j) \in [\![0\,;n]\!]\times[\![0\,;m]\!] \qquad \mathrm{d}(i,j) = \mathrm{lev}(\mathrm{X}[:i],\mathrm{Y}[:j])$$

Les fonctions seront testées au fur et à mesure de leur saisie.

1. Écrire une fonction `lev1(X,Y)` d'arguments `X`, `Y` des chaînes de caractères et qui renvoie la distance de Levenshtein entre `X` et `Y` en réalisant la calcul de manière descendante sans mémoïsation.
2. Écrire une fonction `lev2(X,Y)` d'arguments `X`, `Y` des chaînes de caractères et qui renvoie la distance de Levenshtein entre `X` et `Y` en réalisant la calcul de manière descendante avec mémoïsation.
3. Préciser $\mathrm{d}(i,0)$ pour $i \in [\![0\,;n]\!]$ et $\mathrm{d}(0,j)$ pour $j \in [\![0\,;m]\!]$.
4. Pour $(i,j) \in [\![1\,;n]\!]\times[\![1\,;m]\!]$, déterminer une expression de $\mathrm{d}(i,j)$ fonction de $\mathrm{d}(i-1,j)$, $\mathrm{d}(i,j-1)$, $\mathrm{d}(i-1,j-1)$ et de $\mathrm{X}[i-1]$, $\mathrm{Y}[j-1]$.
5. Compléter fonction `lev3(X,Y)` d'arguments `X`, `Y` des chaînes de caractères pour qu'elle calcule la matrice $\big(\mathrm{d}(i,j)\big)_{(i,j)\in[\![0\,;n]\!]\times[\![0\,;m]\!]}$ de manière ascendante et renvoie $\mathrm{d}(n,m)$.

Exercice 2

Un *alignement* de deux chaînes de caractères X et Y est formé d'un couple de chaînes $\widehat{\mathrm{X}}$ et $\widehat{\mathrm{Y}}$ de même taille, obtenues respectivement à partir des chaînes X et Y avec éventuellement des insertions d'espaces (caractère `'-'`) mais pas sur des positions identiques.

Un *désaccord* dans un alignement est une différence entre deux caractères sur un indice donné.

Un *alignement optimal* de deux chaînes de caractères a et b est un alignement de ces chaînes dont le nombre de désaccords est minimal.

On peut montrer que la distance de Levenshtein entre deux chaînes de caractères est le nombre minimal de désaccords dans un alignement optimal.

Les fonctions seront testées au fur et à mesure de leur saisie.

1. Écrire une fonction `lev_tab(X,Y)` d'arguments `X`, `Y` des chaînes de caractères et qui renvoie la matrice $\big(\mathrm{d}(i,j)\big)_{(i,j)\in[\![0\,;n]\!]\times[\![0\,;m]\!]}$ définie dans l'exercice précédent.
2. Écrire une fonction `lev_align(X,Y)` d'arguments `X`, `Y` des chaînes de caractères et qui renvoie un alignement optimal de ces chaînes en utilisant le résultat de `lev_tab(X,Y)`.
3. Compléter la fonction `lev_align_all(X,Y)` d'arguments `X`, `Y` des chaînes de caractères pour qu'elle renvoie la liste de tous les alignements optimaux de ces chaînes.