

Devoir 4

Remise : le mercredi 4 décembre au *début* de la démo

1. Vous avez accès à une fonction f qui associe à chaque suite w de lettres de longueur un milliard ou moins un nombre réel $f(w)$, $0 \leq f(w) \leq 1$, se voulant un estimé de la fréquence relative à laquelle le mot w risque d'apparaître dans un texte en anglais. Par exemple $f(\text{zahox}) = f(\text{hurzlafootingblu}) < f(\text{handyman}) < f(\text{and})$.

Décrivez en pseudo-code un algorithme de programmation dynamique qui résout :

You have access to a function f that assigns to every sequence w of letters of length a billion or less a real number $f(w)$, $0 \leq f(w) \leq 1$, that estimates the relative frequency of occurrence of w in a body of English text. Describe in pseudo-code a dynamic programming algorithm that exploits f to solve :

SPACEDOUT

DONNÉE: $x_1, x_2, \dots, x_n \in \{a, b, c, d, e, f, g, \dots, y, z\}$

CALCULER: valeur d'une séparation optimale du mot $x_1x_2 \cdots x_n$ obtenue par insertion d'espaces, i.e., formellement,

value of an optimal splitting of $x_1x_2 \cdots x_n$ obtained by the insertion of blanks, i.e., formally,

$$\max \left\{ \sum_{i=1}^k f(w_i) \quad : \quad x_1x_2 \cdots x_n = w_1w_2 \cdots w_k \right\}.$$

Par exemple, en supposant f réaliste, SPACEDOUT sur "helloworld" retournerait sans doute $f(\text{hello}) + f(\text{world})$ et non $f(\text{hell}) + f(\text{ow}) + f(\text{orld})$.

2. Problème 9.22 de BB (articulation partant du sommet 6).
3. Appelons m -chaîne un tableau $T \in \mathbb{N}^{2 \times m}$ tel que pour $1 \leq i \leq m$, $T(1, i) < i$ et $T(2, i) < i$. Sa cible est $\tau(m)$, définie comme suit :
Call an m -chaîne a table $T \in \mathbb{N}^{2 \times m}$ having for each $1 \leq i \leq m$, $T(1, i) < i$ and $T(2, i) < i$. Define the target $\tau(m)$ of such a chaîne as follows :

$$\tau(i) = \begin{cases} 1 & \text{si } i = 0 \\ \tau(T[1, i]) + \tau(T[2, i]) & \text{si } 0 < i \leq m. \end{cases}$$

- (a) Donnez en Python un algorithme qui prend en entrée un tableau $2 \times m$ et qui retourne sa cible si le tableau est une m -chaîne, et -1 sinon.
Give a Python program that takes as input a $2 \times m$ table and returns its target if the table is an m -chaîne, and -1 otherwise.
- (b) Donnez en Python un algorithme à retour arrière qui, étant donné $\tau \in \mathbb{N}^{\geq 2}$, retourne une m -chaîne de cible τ dont le m est minimum.
Give in Python a backtracking algorithm that on input $\tau \in \mathbb{N}^{\geq 2}$ returns an m -chaîne with target τ such that m is minimum.

4. À la semaine prochaine.