Algorithmique du texte

On appelle **texte** une suite finie de caractères (ce que l'on a appelé mot jusqu'à présent). L'algorithmique du texte consiste à résoudre des problèmes sur des textes, qui peuvent en réalité modéliser des informations diverses :

- des textes à proprement parler ;
- une séquence ADN ;
- de la musique ;
- des images...

Par exemple,

- les recherches de similarité, dont notamment :
  - la recherche du plus long sous-mot commun ;
  - la recherche du plus long facteur commun ;
  - la distance d'édition / alignement ;
- la recherche de motif ;
- la compression ;
- l'encodage par facteurs...

1 Rappels

On fixe Σ un ensemble, appelé **alphabet**, dont les élément sont appelés **caractères**. On définit alors l'ensemble $\Sigma*$ des textes sur Σ et celui des textes non vides Σ^+ par :

$$\Sigma^* = \bigcup_{n\in\mathbb{N}} \Sigma^n \qquad \Sigma^+ = \bigcup_{n\in\mathbb{N}^*} \Sigma^n$$

On note ε le texte vide, c'est à dire le seul 0-uplet de Σ^*. On munit alors Σ^* de la **concaténation** définie ainsi :

$$\forall (u = (u_i)_{i\in[\![1,n]\!]}, (v_j)_{j\in[\![1,m]\!]}) \in (\Sigma^*)^2, u.v = u_1u_2...u_{n-1}u_nv_1v_2...v_{m-1}v_m$$

On vérifie alors que $(\Sigma^*, .)$ est un monoïde.

On appelle **sous-texte** d'un texte u de Σ^* toute suite extraite de u. Par ailleurs, on dit que $v \in \Sigma^*$ est un **facteur** de $u = u_1...u_n$ ssi il existe $(i, j) \in [\![1, n]\!]^2$ tel que $v = u_i...u_j$. Dans le cas où $i = 1$, on dit que v est un **préfixe** de u. Si $j = n$, alors c'est un **suffixe** de u.

2 Plus long facteur commun

2.1 Description

Plus long facteur commun	Entrée:	$u \in \Sigma^*$ de longueur n et $v \in \Sigma^*$ de longueur m
	Sortie:	$\max\{\|f\| \| f$ facteur de u et de $v\}$ i.e $\exists(i,j) \in [\![1,n]\!]^2, f = (u_k)_{k\in[\![i,j]\!]}$ ou encore $\exists(p,q) \in [\![1,m]\!]^2, f = (v_k)_{k\in[\![p,q]\!]}$

On abgrègera **Plus long facteur commun** en **PLFC**.

2.2 Résolution

On pose pour $(i,j) \in [\![0,n]\!] \times [\![0,m]\!]$:

$$A_{i,j} = \max\left\{|s| \mid s \text{ est un suffixe de } u_1...u_i, v_1...v_j\right\} \leq \min(i,j)$$

On a immédiatement, pour tout $(i,j) \in [\![1,n]\!] \times [\![1,m]\!]$:

- $i = 0$ ou $j = 0 \to A_{i,j} = 0$
- $A_{i,j} = A_{i-1,j-1} + 1$ si $u_i = v_j$, 0 sinon.

D'ou $\mathbf{PLFC}(u,v) = \max_{(i,j)\in[\![0,n]\!]\times[\![0,m]\!]} A_{i,j}$

3 Recherche de motif

3.1 Définitions

Trouve motif	Entrée:	$t \in \Sigma^*$ un texte de longueur n, $x \in \Sigma^*$ un motif de longueur m (avec $m \leq n$)
	Sortie:	$\underbrace{\{i \in [\![1,n]\!] \mid (t_{i+k})_{k\in[\![0,m[\![=x}\}}_{\text{l'ensemble des indices de début des occurences de } x \text{ dans } t}$

Exercice 1: Proposer un algorithme de résolution naïve pour ce problème.

3.2 Algorithme de Rabin-Karp

3.3 Algorithme de Boyer-Moore-Horspool

3.4 Algorithme de Boyer-Moore