M1 Bio-Info, Info et Math-Info
Algorithmique du texte
HMIN121M

Année scolaire 2018-2019

TD : Arbre des suffixes

Sèverine Bérard - ISE-M, Faculté des Sciences, Université de Montpellier

Severine.Berard@umontpellier.fr

Soit $\mathcal{T} = (X, U)$ un arbre des suffixes et $\mathcal{A}$ un alphabet fini.

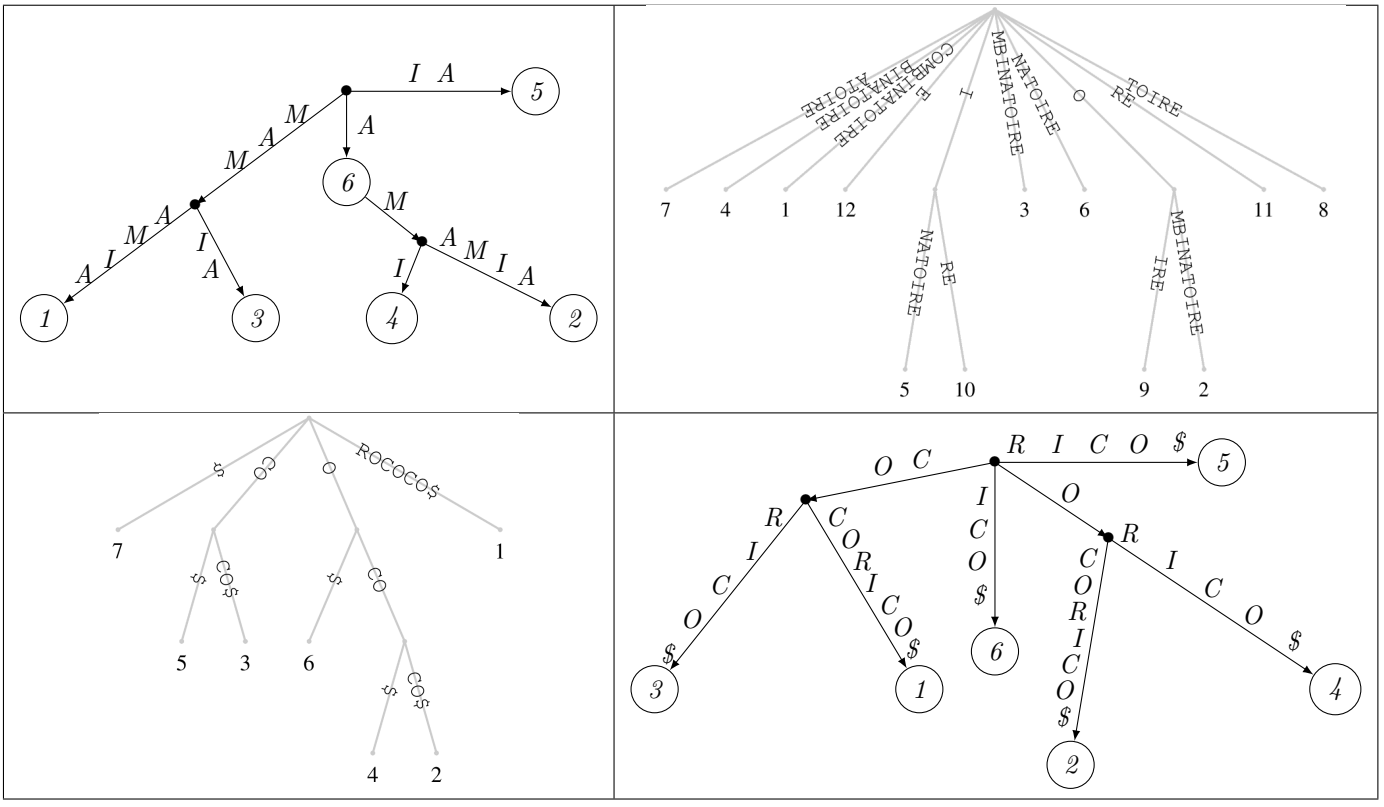
Exercice 1

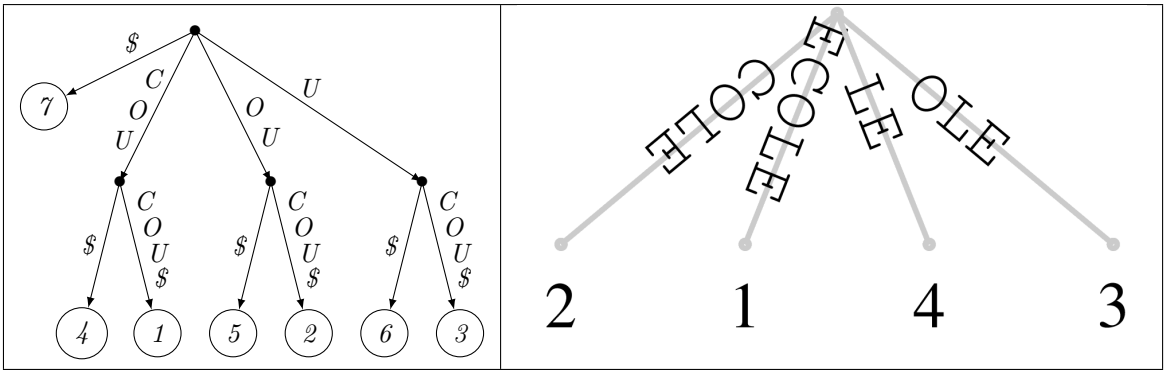
Donnez les ensembles de départ et d'arrivée des applications suivantes :

1. **e** *l'application étiquette qui à tout sommet et à tout chemin de* $\mathcal{T}$ *associe son étiquette ;*
2. **p** *qui à tout nœud de* $\mathcal{T}$ *associe sa profondeur ;*
3. $\mathbf{p_c}$ *qui à tout nœud de* $\mathcal{T}$ *associe sa profondeur de chaîne ;*
4. **ch** *qui à toute chaîne* a *de* $\mathcal{A}^*$ *associe le nœud ou l'arc sur lequel se finit le chemin associé à* a.

Exercice 2

Parmi les arborescences suivantes, indiquez si elles sont ou pas des arbres des suffixes. Si oui, précisez de quelle séquence, sinon indiquez le ou les élément(s) non conforme(s) à la définition vue en cours.





Exercice 3

Pour le dernier arbre des suffixes valide de l'exercice précédent, donnez des exemples de valeur pour les applications que vous avez définies à l'exercice 1.

Exercice 4

Construire un arbre des suffixes pour la chaîne $S = ABBAABA$. Est-ce le seul possible ? Justifiez.

Exercice 5 (Répétitions)

Écrire un algorithme permettant de détecter toutes les répétitions de taille $k > 0$ dans une séquence S de longueur $n > k$.

Même question pour détecter les répétitions de taille maximale.

Exercice 6

Proposez une solution efficace pour réduire la taille des étiquettes des arcs dans l'arbre des suffixes.

Exercice 7 (Exact set matching problem)

On s'intéresse à nouveau au problème de recherche exacte d'un ensemble de motifs $\mathcal{P}$ de longueurs cumulées m dans un texte T de longueur n.

1. *Proposez un algorithme utilisant les arbres des suffixes (on supposera que l'on peut construire un arbre des suffixes d'une séquence S en temps linéaire en la longueur de S) ;*
2. *Calculez les complexités en temps et en espace de votre solution ;*
3. *Comparez votre solution avec l'algorithme de Aho-Corasick vu en cours.*

Exercice 8 (Longuest Common Substring)

Soient 2 chaînes de caractères S_1 et S_2, proposez un algorithme permettant de donner une plus longue sous-chaîne (=facteur) commune aux 2 séquences. Calculez ses complexités en temps et en espace.