

.1 Mots

1. **Désaccentuation** Écrire un programme qui, étant donnée une chaîne de caractères (stockée dans un tableau), « désaccentue » la chaîne : tous les caractères accentués sont remplacés par le caractère correspondant sans diacritique.
2. **Sous-mot** Écrire un programme qui vérifie qu'un mot est un sous-mot d'un autre.
3. **Génération de facteurs** Écrire un programme qui génère tous les facteurs d'un mot donné (sans se préoccuper de proposer plusieurs fois le même).
4. **KMP** Implémenter et tester l'algorithme de Knuth, Morris et Pratt. On procédera de la manière suivante.
 - Soit $x[i]$ le $i^{\text{ème}}$ caractère du facteur recherché
 - On appelle *frontière interne* du sous-facteur $x[0..i]$ le préfixe v de x tel que :
 - v soit égal à un suffixe de $x[0..i-1]$
 - v soit suivi d'un caractère différent de $x[i]$
 - On définit $\text{kmpNext}[i]$ ($0 < i \leq m$) comme :
 - la frontière interne de $x[0..i]$ si elle existe
 - -1 sinonÉcrire une fonction qui calcule le contenu de `kmpNext` pour un facteur donné.
 - Écrivez l'algorithme général KMP. Vous devrez utiliser le contenu de `kmpNext` et de manière plus générale vous inspirer de l'algorithme utilisé pour construire ce tableau.
 - Faites “tourner” l'algorithme sur un exemple de votre choix.

.2 Arbres Binaires de Recherche

1. Proposer un algorithme récursif de recherche dans un arbre binaire
2. Soit un ABR. Décrire l'opération d'insertion d'un noeud dans l'arbre.
3. Soit la suite d'entiers $s = 11, 15, 5, 2, 3, 9, 17, 21, 22, 13, 19, 4, 12$
 - (a) Construire un arbre binaire de recherche associé à s (par exemple en prenant les éléments dans l'ordre donné)
 - (b) Dessiner le chemin de recherche des nombres 5, 19 et 8 dans l'arbre
 - (c) Donner le résultat de la suppression de la racine. Supprimer ensuite 15 et 17.
 - (d) Quel parcours de cet arbre permet de trier les nombres dans l'ordre croissant ?
4. Donner tous les arbres binaires de recherche possibles formés des éléments 1,2,3,4. Quels sont ceux de profondeur minimale et pourquoi sont-ils préférables ?
5. Sous-arbres :
 - (a) Écrire une fonction qui teste l'égalité de deux arbres binaires étiquetés
 - (b) Un arbre binaire S figure dans un arbre binaire T s'il lui est égal ou s'il est égal à sous-arbre de T . Écrire une fonction récursive $figure(S, T)$ qui indique si S figure dans T ; si c'est le cas, la fonction renvoie l'étiquette du noeud racine du sous-arbre de T qui est égal à S et NULL sinon.
6. On définit une fonction entière ϕ sur l'ensemble des arbres binaires. On appelle $g(A)$ et $d(A)$ respectivement le sous-arbre gauche et droit de A .

$$\phi(A) = \begin{cases} 0 & \text{si } A \text{ est l'arbre vide} \\ \max(\phi(g(A)), \phi(d(A))) & \text{si } \phi(g(A)) \neq \phi(d(A)) \\ \phi(d(A)) + 1 & \text{si } \phi(g(A)) = \phi(d(A)) \end{cases}$$

Cette fonction ϕ permet d'associer à tout arbre binaire un **nombre de Strahler**.

- (a) Quel est le nombre de Strahler d'un arbre binaire complet de hauteur n ? Celui d'un arbre binaire dégénéré de hauteur n ?
- (b) Donner une procédure récursive basée sur le parcours vu en cours qui calcule le nombre de Strahler d'un arbre binaire.