Algorithmique – Programmation Objet – Python
TD n° 12
Arbre des suffixes

Licence Informatique 2ème année
Université de Nice-Sophia Antipolis

Introduction

Un arbre des suffixes est une structure de données contenant tous les suffixes d'un texte, qui trouve son utilisation dans l'indexation des textes pour la recherche de motifs.

Soit T une chaîne de caractères (que nous appelerons *le texte*) dont on veut construire l'arbre des suffixes. L'arbre a les propriétés suivantes :

- Chaque arête est étiquetée par un caractère de T.
- Chaque arête partant du même nœud est étiquetée par un caractère différent.
- Chaque nœud de l'arbre contient un indice qui correspond à la position de début de la première occurrence dans T de la sous-chaîne correspondant à son chemin.
- Chaque chemin de la racine à une feuille correspond à un suffixe de T qui débute à la position marqée dans la feuille.

Cette structure permet de rechercher un motif m dans un texte de longueur n en temps $O(\|m\|)$, c'est-à-dire un temps qui dépend uniquement de la longueur du motif.

1 Conception d'une classe ArbreSuff

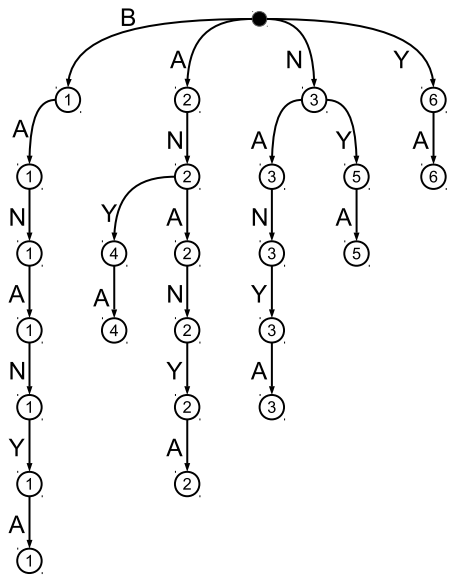
Concevoir une classe ArbreSuff qui encapsule un arbre des suffixes tel qu'il vient d'être décrit. Détaillez les attributs que cette classe devra contenir.

2 Construction de l'arbre des suffixes

Esko Ukkonen, de l'Université de Helsinki, a proposé un algorithme de construction de l'arbre des suffixes d'un texte T *en ligne*, c'est-à-dire, qui considère les caractères du texte un à un, du premier au dernier, sans jamais devoir revenir en arrière, et, à chaque pas, il dispose de l'arbre des suffixes complet pour la partie de texte déjà couverte. L'algorithme se base sur l'observation que l'arbre des suffixes de la chaîne $T^i = c_1 \cdots c_i$ peut être obtenu à partir de l'arbre des suffixes de la chaîne $T^{i-1} = c_1 \cdots c_{i-1}$ par l'ajout du caractère c_i à la fin de chacun des suffixes de T^{i-1}. Les suffixes de T peuvent donc être obtenus en étendant d'abord les suffixes de T^0 pour obtenir les suffixes de T^1 et ainsi de suite, jusqu'à ce que les suffixes de $T = T^n$ sont obtenus à partir des suffixes de T^{n-1}.

Exploiter cette idée pour réaliser le constructeur ArbreSuff(T) de la classe ArbreSuff, qui prend un texte T en entrée et construit l'arbre des suffixes correspondant. Quelques suggestions :

- On se servira de deux listes de nœuds *ouverts*, c'est-à-dire correspondants aux suffixes de l'itération : la liste des nœuds ouverts pour l'itération courante et celle des nœuds ouverts pour l'itération suivante. La racine fait toujours partie des deux listes.
- Pour chaque caractère du texte c_i, on parcourt la liste des nœuds ouverts pour l'itération courante et on vérifie s'il existe déjà une arête sortante étiquetée par c_i. Si elle n'existe pas, on ajoute un nouveau nœud fils, relié au parent par une arête étiquetée par c_i. Quoi qu'il en soit, on ajoute le nœud fils aux nœuds ouvert pour la prochaine itération.
- Lorsqu'un nouveau nœud est ajouté à l'arbre, l'indice qu'il va contenir est donné par i, l'indice du caractère courant, moins la profondeur de son parent.
- Utiliser le texte $T =$ "`BANANYA`" pour tester l'algorithme : l'arbre construit devrait ressembler à celui-ci:



3 Recherche d'un motif

La recherche d'un motif est relativement simple. En partant de la racine, on suit l'arête étiquetée par le premier caractère du motif, puis à partir du nœud où on est arrivés, on suit l'arête étiquetée par le prochain caractère du motif, et ainsi de suite. On s'arrête

- soit parce qu'on a rejoint le dernier caractère du motif, et alors on trouve dans le nœud courant l'indice de la première occurrence du motif dans la chaîne de caractères,
- soit parce qu'il n'y a pas d'arête étiquetée par le prochain caractère du motif partant du nœud courant, et alors on sait que le motif recherché n'existe pas dans la chaîne de caractères.

Écrire la méthode RECHERCHER(m), qui prend une chaîne de caractères m en entrée (le motif) et renvoie l'indice de la première occurrence de m dans le texte dont l'arbre des suffixes est une représentation, -1 s'il n'y a pas d'occurrence de m dans le texte.