EPU GB4-BIMB **TD d'Algorithmique pour la biologie n.2** février 2024

String Matching

Exercice 1 : (**Algorithme de Rabin-Karp**)

Cet exercice est dédié à l'implémentation de l'algorithme de Rabin-Karp. Pour simplifier, nous considérons des chaînes de caractères écrites sur l'alphabet `alphabet = ['A','T','C','G']`.

1. Définir en Python un dictionnaire qui représentera le codage des lettres de l'alphabet.
2. Implémentez l'algorithme de Rabin-Karp. Les paramètres de la fonction sont au nombre de 4 : le texte dans lequel on recherche un motif, le motif recherché, la taille de l'alphabet, et enfin le nombre q utilisé pour calculer le modulo.
3. Comment expérimenter que plus q est petit, plus il y aura d'appels à la phase de recherche lettre à lettre ?

Exercice 2 : (**Recherche d'un motif à l'aide d'un automate, A RENDRE**)

1. Construisez à la main l'automate pour le motif `AAAT`.
2. Comment peut-on représenter un automate en Python ? Représentez en Python l'automate de la question précédente.
3. Implémentez l'algorithme qui permet de parcourir un texte à l'aide d'un automate.
4. Testez votre implémentation avec l'automate obtenu en 2.
5. Ecrivez une fonction qui teste si un mot est un suffixe d'un autre.
6. Implémentez l'algorithme de construction de l'automate associé à un motif.
7. Quelle est la commande pour trouver les occurrences d'un mot donné dans un texte donné ?

Exercice 3 : (**Algorithme de Knuth-Morris-Pratt**)

1. Ecrivez une fonction `calculFonctionPrefixe` qui prend en argument un mot et renvoie un dictionnaire représentant la fonction $\pi()$ de l'algorithme de Knuth-Morris-Pratt. Si le mot recherché P est de taille m, alors la fonction $\pi()$ prend en entrée un entier $q \in [1, m]$ (correspondant à la taille du plus grand préfixe lu jusqu'à présent) et renvoie $\pi(q)$ qui est la taille du plus grand suffixe **strict** de $P[0, q[$, qui soit aussi préfixe de P.
2. Ecrivez ensuite une procédure qui permet de recherche toutes les occurences d'un mot dans un texte en utilisant l'algorithme de Knuth-Morris-Pratt.