Recherche optimisée de motif dans un texte

Introduction : De nombreuses pages de texte sont disponibles sous format numérisé et cela ouvre de nombreuses possibilités. Parmi elles, la possibilité de chercher un mot ou un motif dans un texte. Cette fonctionnalité est implémentée dans tous les logiciels de base (Ctrl-F) et nous allons voir comment on peut la programmer, et surtout vue sa fréquence d'utilisation comment la programmer *efficacement*.

I – Problème

On se donne le texte suivant[1]:

ttcagttgtgaatgaatggacgtgccaaatagacgtgccgccgccgctcgattcgcactttgctttcggttttgccgtcgtttcacgcgtttagttccg
ttcggttcattcccagttctttgctttcggttttgccgtcgtttcacgcgtttagttccgttcggttcattcccagttcttgtagaaatattaaaataaattcct

On y cherche le motif **ttagttccgt**

L'objectif du chapitre sera d'écrire un programme python qui :

- renvoie la première position du motif dans le texte s'il y est présent ;
- renvoie -1 sinon.

II – Recherche naïve

Comment pourrait-on procéder pour résoudre le problème ? Expliquer le principe de l'algorithme. Quel serait sa complexité ?

1 Les algorithmes de manipulation de texte aux seules quatre lettres A, T, C, G sont très utilisés dans le domaine de la génétique. La discipline associée est la **bioinformatique**.

III – Recherche optimisée – Vers l'algorithme de Boyer-Moore

L'idée majeure de l'algorithme de Boyer-Moore est de **ne lire que les caractère du texte qui ont une chance de figurer dans le motif**. Pour cela, on lit le texte depuis le début mais **on lit le motif depuis la fin**.

1°) Quelle est la longueur du motif **ttagttccgt** ?

2°) Si le texte commence par le motif, quel doit être la valeur du dixième caractère du texte ? Est-ce le cas ?

3°) En supposant que le caractère lu dans le texte fait partie du motif, à quelle position se trouverait son dernier caractère ? Quelle en serait la valeur ?

4°) La réponse à la question précédente dépend-elle de la position où l'on se trouve dans le texte ? Comment peut-on exploiter cela ? Avec quelle structure de donnée ?

5°) En déduire un algorithme optimisé pour la recherche de motif. Quels sont ses avantages par rapport à une recherche naïve ? L'algorithme sera-t-il plus intéressant pour un motif long ou pour un motif court ?