

Exercice 1 (14points)
Le but de cet exercice est de trouver le plus long mot commun à deux textes donnés T1 et T2 sur un alphabet. Un mot d'un texte est dans ce contexte toute suite de caractères consécutifs du texte. Nous notons n1 et n2 respectivement les longueurs de T1 et T2.
Pour cela, nous envisageons l'utilisation de tous les algorithmes de recherche exacte vus en cours, en les comparant selon leur complexité en temps.

Utilisation d'algorithmes de recherche par fenêtre glissante : AlgoZ, KMP(Knuth-Moris-Pratt), BM(Boyer-Moore)

L'utilisation de ces algorithmes est similaire, à savoir : on considère tous les mots de l'un des textes et on les recherche de manière exacte dans l'autre texte à l'aide d'un algorithme X2 fAlgoZ;KMP;BMg, en gardant en mémoire le mot le plus long qui a été trouvé. plus ou moins efficace.

Q1. Ecrire en pseudo-code un algorithme pour calculer le plus long mot commun à T1 et à T2 selon l'approche indiquée ci-dessus.
Appelons Rechercher1-X(T1; T2) cet algorithme, en fonction de l'algorithme de recherche X qu'il utilise (et qui doit être appelé par son nom, comme une fonction ou procédure).
La variante Rechercher1-AlgoZ utilise donc AlgoZ pour la recherche exacte, la variante Rechercher1-KMP utilise KMP et la variante Rechercher1-BM utilise BM.

Q2. Calculer - en justifiant - la complexité de chacune de ces variantes comme suit :
(a) dans le pire des cas, en fonction de n1 et de n2
(b) lorsque le mot le plus long (c'est-à-dire la solution) a une longueur de L (avec $0 \leq L \leq \min(n_1, n_2)$).
Dans ce cas, la complexité de l'algorithme peut dépendre aussi de L.
A la fin de ce calcul, vous devez remplir le tableau suivant :

	Rechercher1-AlgoZ	Rechercher1-KMP	Rechercher1-BM
Au pire			
Lorsque la solution est de taille L			

Q3. Appliquer l'algorithme Rechercher1-BM sur l'exemple suivant (tous les paramètres nécessaires à l'application de l'algorithme BM seront calculés rapidement sans détailler) :
T1 : bourbourgeois
T2 : bourgeon
Vous donnerez les détails de l'exécution, sauf ceux pour le calcul des paramètres (Note. Un bourbourgeois est un habitant de la ville de Bourbourg, dans le Nord.)

Utilisation des algorithmes de recherche exacte à base d'index : tableau et arbre des suffixes
A priori, ces structures devraient permettre de résoudre le problème sans chercher successivement tous les mots d'un texte dans l'autre texte.

Q4. Donner, sous la forme d'un algorithme dont on ne décrira que les étapes principales, une méthode pour résoudre le problème de recherche du plus long mot commun à deux textes utilisant la structure de tableau des suffixes. Soit Rechercher2-TS(T1; T2) le nom de cette méthode.
Q5. Donner, sous la forme d'un algorithme dont on ne décrira que les étapes principales, une méthode pour résoudre le même problème utilisant la structure d'arbre des suffixes. Soit Rechercher2-AS(T1; T2) le nom de cette méthode.

Q6. Calculer - en la justifiant - la complexité de chacune de ces variantes comme suit :

(a) dans le pire des cas, en fonction de n_1 et de n_2

(b) lorsque le mot le plus long (c'est à dire la solution) a une longueur de L (avec $0 \leq L \leq \min(n_1; n_2)$).

Dans ce cas, la complexité de l'algorithme doit dépendre aussi de L .

A la fin de ce calcul, vous devez remplir le tableau suivant :

	Rechercher2-TS	Rechercher2-AS
Au pire		
Lorsque la solution est de taille L		

Q7. Appliquer l'algorithme le plus efficace parmi Rechercher2-TS et Rechercher2-AS sur l'exemple suivant :

T_1 : bourbourgeois

T_2 : bourgeon

Vous donnerez les détails de l'exécution (sauf la construction de TS ou AS).

Exercice 2 (6 points)

Q8. Pourquoi l'approche dite « standard » ou encore « directe » à partir de corpus comparables est-elle peu appropriée pour l'alignement de termes complexes ? (3 points)

Q9. Après avoir rappelé le fonctionnement de la mesure de comparabilité de Li et Gaussier (2010), pouvez-vous expliquer le résultat attendu de celle-ci sur un corpus parallèle (3 points) ?