

Contrôle continu du 19 décembre 2018

**Durée : 80min (pour la partie Recherche de motif du sujet)**

Documents de CM/TD autorisés. Présentation, clarté et orthographe seront pris en compte dans la note finale. Il est également important de bien justifier toutes vos réponses.

**Exercice 1.**

[ ? ? ? ? min, 14pts]

Le but de cet exercice est de trouver le plus long mot commun à deux textes données  $T_1$  et  $T_2$  sur un alphabet  $\Sigma$ . Un *mot* d'un texte est dans ce contexte toute suite de caractères consécutifs du texte. Nous notons  $n_1$  et  $n_2$  respectivement les longueurs de  $T_1$  et  $T_2$ .

Pour cela, nous envisageons l'utilisation de tous les algorithmes de recherche exacte vus en cours.

**Utilisation d'algorithmes de recherche par fenêtre glissante : AlgoZ, KMP<sup>1</sup>, BM<sup>2</sup>**

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  $X \in \{\text{AlgoZ}, \text{KMP}, \text{BM}\}$ , en gardant en mémoire le mot le plus long qui a été trouvé.

Q1. Ecrire en pseudo-code un premier algorithme, pas encore optimisé, pour calculer le plus long mot commun à  $T_1$  et à  $T_2$  selon l'approche indiquée ci-dessus. Appelons  $\text{Cherche1-X}(T_1, T_2)$  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  $\text{Cherche1-AlgoZ}$  utilise donc AlgoZ pour la recherche exacte, la variante  $\text{Cherche1-KMP}$  utilise KMP et la variante  $\text{Cherche1-BM}$  utilise BM.

Q2. Calculer la complexité de chacune de ces variantes comme suit :

1. dans le pire des cas, en fonction de  $n_1$  et de  $n_2$
2. lorsque le mot le plus long a une longueur de  $L$  (avec  $0 \leq L \leq \max\{n_1, n_2\}$ ). 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 :

	Cherche1-AlgoZ	Cherche1-KMP	Cherche1-BM
Au pire			
Lorsque la solution est de taille $L$			

Il s'agit par la suite d'améliorer cet algorithme.

Q3. Comment les complexités que vous avez calculées dépendent-elles de  $n_1$  et de  $n_2$  ? De quelle manière peut-on améliorer l'algorithme en prenant en compte vos observations ?

Etant donné que l'on cherche les mots communs les plus longs, une deuxième approche serait de considérer les mots du premier texte par ordre décroissant des tailles : d'abord tous les mots de taille maximum, ensuite tous les mots qui ont seulement un caractère de moins, ensuite tous les mots qui ont encore un caractère de moins, etc. pour s'arrêter dès qu'on a trouvé un mot commun (s'il y en a un).

Q4. Ecrire un deuxième algorithme pour calculer le plus long mot commun à  $T_1$  et à  $T_2$  selon l'approche indiquée ci-dessus. Appelons  $\text{Cherche2-X}(T_1, T_2)$  cet algorithme, en fonction de l'algorithme de recherche  $X$  qu'il utilise. La variante  $\text{Cherche2-AlgoZ}$  utilise donc AlgoZ pour la recherche exacte, la variante  $\text{Cherche2-KMP}$  utilise KMP et la variante  $\text{Cherche2-BM}$  utilise BM.

Q5. Calculer la complexité de chacune de ces variantes comme suit :

1. dans le pire des cas, en fonction de  $n_1$  et de  $n_2$
2. lorsque le mot le plus long a une longueur de  $L$  (avec  $0 \leq L \leq \max\{n_1, n_2\}$ ). Dans ce cas, la complexité de l'algorithme doit dépendre aussi de  $L$ .

---

1. Knuth-Morris-Pratt  
2. Boyer-Moore

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

	Cherche2-AlgoZ	Cherche2-KMP	Cherche2-BM
Au pire			
Lorsque la solution est de taille $L$			

Q6. Quel algorithme est le plus efficace parmi les six variantes proposées jusqu'ici ? Pourquoi ?

Q7. Appliquer l'algorithme le plus efficace parmi Cherche1-BM et Cherche2-BM sur l'exemple suivant (tous les paramètres nécessaires à l'application de l'algorithme seront calculés à la main) :

$T_1$  : bourbourgeois<sup>3</sup>

$T_2$  : bourgeon

Vous donnerez les détails de l'exécution.

#### 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.

Q8. Donner, sous la forme d'un algorithme dont on ne décrira que les étapes principales, une méthode pour résoudre le problème utilisant la structure de tableau des suffixes. Soit Cherche3-TS( $T_1, T_2$ ) le nom de cette méthode.

Q9. Donner, sous la forme d'un algorithme dont on ne décrira que les étapes principales, une méthode pour résoudre le problème utilisant la structure d'arbre des suffixes. Soit Cherche3-AS( $T_1, T_2$ ) le nom de cette méthode.

Q10. Calculer la complexité de chacune de ces variantes comme suit :

1. dans le pire des cas, en fonction de  $n_1$  et de  $n_2$
2. lorsque le mot le plus long a une longueur de  $L$  (avec  $0 \leq L \leq \max\{n_1, n_2\}$ ). 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 :

	Cherche3-TS	Cherche3-AS
Au pire		
Lorsque la solution est de taille $L$		

Q11. Appliquer l'algorithme le plus efficace parmi Cherche3-TS et Cherche3-AS sur l'exemple suivant :

$T_1$  : bourbourgeois<sup>4</sup>

$T_2$  : bourgeon

Vous donnerez les détails de l'exécution.

Q12. Quel algorithme, parmi les 8 variantes proposées, est le plus efficace ? Justifier.

---

3. Habitant de la ville de Bourbourg, dans le Nord

4. Habitant de la ville de Bourbourg, dans le Nord