

Algorithmique
DST n°2 & Examen
Aucun document autorisé
Durée : 2 heures

1. On veut réaliser une fonction qui, étant donnée une pile contenant n éléments $(u_1 u_2 \dots u_n)$, renvoie la pile *inversée* : la valeur u_1 est au sommet, la valeur u_n est au fond de la pile $(u_n u_{n-1} \dots u_1)$.
 - (a) Proposer une version qui utilise les primitives de gestion de pile. De quelle structure de données a-t-on besoin ?
 - (b) En supposant que la pile est implémentée avec un tableau (implémentation contiguë), implémenter l'inversion comme une primitive.
 - (c) Comparer la complexité des deux méthodes.

2. Etant donné un alphabet X (on suppose qu'il a 26 symboles), on suppose défini un arbre⁶ dont chaque nœud a 26 fils, de sorte que tout mot sur X^* correspond exactement à un nœud dans l'arbre.
 - (a) On dira qu'un mot u précède un autre mot v (ordre lexicographique) si le mot u est placé dans l'arbre de telle sorte qu'il serait rencontré avant v dans un parcours en profondeur d'abord.
 En supposant définies les primitives suivantes :

<i>boolean</i>	<code>domine(X, Y)</code>
<i>nœud</i>	<code>premier_ancetre_commun(X, Y)</code>
<i>list de nœuds</i>	<code>chemin(X, Y)</code>
<i>nœud</i>	<code>element_chemin(X, Y, i)</code>

 définir une fonction booléenne `avant(X, Y)` qui renvoie `True` si X est avant Y dans l'ordre lexicographique.
 - (b) En supposant définie la primitive `liste_fils(X)` (et les primitives habituelles de manipulation de liste), proposer une implémentation récursive de la primitive `domine(X, Y)`.
 - (c) On suppose maintenant qu'un nœud est représenté par une chaîne de caractères (sur X). En supposant définies les opérations habituelles sur les chaînes de caractères, définir chacune des primitives :

<code>fils(X)</code>
<code>frere(X)</code>
<code>existe_frere(X)</code>
<code>pere(X)</code>
<code>profondeur(X)</code>
 - (d) Avec les mêmes hypothèses que celles de la question précédente, proposer une implémentation pour la primitive `premier_ancetre_commun(X, Y)` (qui doit renvoyer un nœud, c'est-à-dire un mot).
 À quoi correspond la primitive `chemin(X, Y)` ?
 - (e) En laissant de côté toutes les questions précédentes, proposer une implémentation de la fonction `avant(X, Y)` qui prend en argument deux chaînes de caractères et renvoie la première dans l'ordre lexicographique.
 Commenter.

3. On suppose que l'on représente un multi-ensemble \mathcal{E} d'entiers entre 0 et 100 par un tableau de 100 valeurs tel que :

<code>t[i] == 0</code>	si $i \notin \mathcal{E}$
<code>t[i] == a</code>	($a \neq 0$) si i compte a occurrences dans \mathcal{E}

 Proposer une fonction qui implémente l'*intersection* entre deux multi-ensembles ainsi codés : prenant en argument deux tableaux représentant des multi-ensemble, elle renvoie un tableau représentant l'intersection.

⁶potentiellement infini, mais on peut le supposer de hauteur bornée dans toute la suite de l'exercice.