

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

1. On suppose que l'on dispose des primitives de manipulation de liste suivantes, avec les définitions habituelles :

```

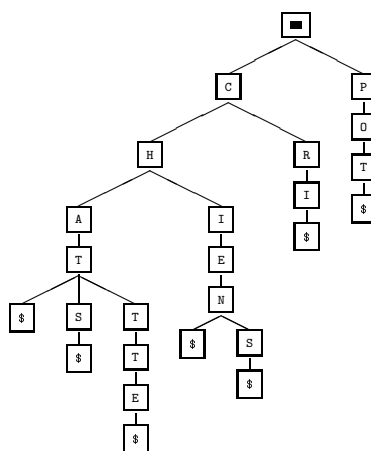
void Insérer(liste l, int p, E x)
void Supprimer(liste l, int p)
E Élément(liste l, int p)
int Longueur(liste l)
    
```

On suppose que l'on dispose d'une liste donnée, \mathcal{L} .

- (a) Écrire un algorithme qui affiche tous les éléments de la liste \mathcal{L} à l'écran.
 - (b) Modifier ce programme pour que les éléments qui apparaissent plusieurs fois ne soient pas affichés (en d'autres termes, ne seront affichés à l'écran que les éléments qui apparaissent une seule fois dans la liste). Quelle est la complexité de cet algorithme ?
 - (c) Écrire un algorithme qui supprime les doublons de la liste, c'est-à-dire qui ne laisse qu'une occurrence de chaque élément.
2. [Trois questions au choix. Les questions sont indépendantes] On suppose qu'on veut stocker un dictionnaire sous la forme d'un arbre, à la manière illustrée à la figure suivante. On suppose que les primitives suivantes servent à définir l'arbre.

```

noeud racine()
boolean existe_fils_lettre(noeud n, char c)
noeud fils_lettre(noeud n, char c)
    
```



- (a) Donner un algorithme qui recherche un mot (donné sous forme d'un tableau de caractères) dans un tel arbre. On demande une version récursive, et une version itérative.
- (b) Proposer une structure de données (contiguë ou chaînée) qui permette de stocker de façon efficace un tel arbre. Indiquez sommairement la complexité de la primitive *existe_fils_lettre()*
- (c) On souhaite maintenant afficher dans l'ordre alphabétique tous les mots de ce dictionnaire. Est-ce que les primitives ci-dessus sont les plus appropriées ? Donner l'algorithme d'affichage en utilisant les primitives de votre choix.
- (d) On suppose définie une méthode supplémentaire, *ajoute_noeud_lettre()* qui prend en paramètre une lettre et l'ajoute comme fils au noeud argument. Cette méthode n'est définie que si le noeud n'a pas déjà cette lettre comme fils. Proposer un algorithme qui ajoute un mot dans le dictionnaire (par exemple *chaton*).
- (e) Expliquer en langage naturel l'algorithme à mettre en œuvre pour supprimer un mot dans un tel dictionnaire (ici, par exemple, le mot *chats*).