



Travaux Dirigés d'informatique n°6
Algorithmique
Arbres binaires fils/frère et lexicographie.

Introduction

On rappelle la structure des arbres fils/frère:

```
typedef struct noeud
{
    struct noeud* frere;
    struct noeud* fils;
    char lettre;
} Noeud, *Arbre;
```

Exercice 1 Insertion.

1. Dessiner l'arbre obtenu après insertion des mots suivants:

- bavard
- bavette
- bavaroise
- billard
- biere
- bar

2. écrire la fonction `int insertion(Arbre *A, char *mot)` qui insère un mot dans l'arbre A, la fonction renverra 1 si l'arbre a bien été modifié, 0 sinon.

Exercice 2 Recherche d'un mot.

1. Sur l'exemple précédent, décrivez la recherche du mot *biere*, celui du mot *bille* et du mot *bave*.

2. écrire la fonction `int recherche(Arbre A, char *mot)` qui renvoie 1 si le mot est bien dans l'arbre A, et 0 sinon.

Exercice 3 Affichage.

1. Sur le même exemple, décrivez l'algorithme d'affichage pour le parcours fils, suivi du frère.

2. écrire la fonction `void affichage(Arbre A)` qui affiche les mots de A. Vous utiliserez une fonction auxiliaire `void affichageaux(Arbre A, char buffer[], int indice)`, où `buffer` est un tableau de caractères, de taille une constante `MAX` prédéfinie et `indice` correspond à la première case vide du tableau `buffer`.