TP n° 3 : Arbres Binaires de Recherche

2015-2016

Objectif *:*

- Implémenter une structure hiérarchique à l'aide d'un **A**rbre **B**inaire de **R**echerche **ABR**

1. Construction d'un ABR

On considère un **ABR** où chaque nœud contient un entier (la clé). La déclaration de l'arbre est :

```
typedef struct _arbre {
      int cle;
      struct _arbre *gauche;      struct _arbre *droit;
} arbre;
```

a. Ecrire un programme qui construit l'arbre **ABR** à partir du tableau

`t[]={55,34,49,20,38,58,10,50,25,22,24};`

b. Afficher l'arbre en l'explorant et supprimer ensuite les valeurs `55` puis `49` puis `66`
Vous aurez à écrire les fonctions suivantes :

```
arbre * ajouter(int el, arbre *r)  // ajouter un élément dans l'arbre
void  explorer(arbre *r)              // explorer l'arbre en in ordre
void supprimer(int el,arbre **r) // supprimer un élément de l'arbre
```

2. ABR avec chaînes de caractères comme clé

Nous voulons afficher tous les mots d'un texte avec leur nombre d'apparitions. Nous représentons le texte par un **A**rbre **B**inaire de **R**echerche (**ABR**) où chaque nœud représentant un mot contient deux champs : le **mot** proprement dit et son **nombre d'apparitions**.

```
   char mot[80] ; // le mot lui-même (la clé)
   int  occ ;   // le nombre d'occurrences  du mot dans le texte
```

a. Ecrire un programme de test qui considère que le texte *est lu au clavier* **un mot par ligne**, et afficher les mots en ordre lexicographique.

b. Changer le programme pour qu'il puisse *lire au clavier* **plusieurs mots par ligne**. Les mots sont séparés par un `espace`, une `tabulation` ou un `retour` à la ligne.

Vous pouvez utiliser la fonction `lire_mot()` suivante (`SKIP` étant l'instruction vide)

```
int lire_mot(char *mot, int maxm){
      char c; char *m=mot;
      while( !isalpha(c=getchar())  && (c!= EOF) )  SKIP;
      if(c == EOF) return EOF;
      *m=c; m++;
      for(SKIP ; isalnum( c=getchar() ) && (--maxm>0) ; SKIP){
          *m=c; m++; }
      *m ='\0';
      return c;
}
```

c. Maintenant le programme doit **lire les mots à partir d'un fichier texte**. Vous devez modifier pour cela la fonction `lire_mot()`.

```
int flire_mot(FILE *f, char *mot, int maxm){… }
```

Annexe

ABR ?

- Un **ABR** est un arbre binaire dont la clé d'un nœud est supérieure à toutes celles de son sous arbre gauche inférieure à toutes celles de son sous arbre droit