Travaux Pratiques d'algorithmique n°6

Cours d'Informatique de Deuxième Année

—DEUG Sciences S3-MI—

Manipulation d'ensembles triés

Au cours de cette séance, nous reprenons les structures du tp précédent. Nous travaillons maintenant sur des ensembles triés. Nous implémentons tout d'abord les fonctions de base sur les ensembles triés, puis nous traitons l'union et l'intersection de deux ensembles triés et l'inclusion d'un ensemble trié dans un autre.

Nous rappelons que :

- le type `Produit` utilisé durant ce TP est défini de la façon suivante (où `MAXSTR` est la taille maximale d'une chaîne de caractères) :

```
struct produit
{
   char          nom[MAXSTR];
   char          fournisseur[MAXSTR];
   double        prix;
};
typedef struct produit Produit;
```

- le type `Ensemble` utilisé durant ce TP est défini de la façon suivante (où `MAXENS` est la taille maximale de produit que peut contenir un ensemble) :

```
struct ensemble
{
   unsigned int taille;
   Produit      tab[MAXENS];
};
typedef struct ensemble Ensemble;
```

Vous aurez besoin des fonctions de bases implémentées lors du TP précédent.

▸ **Exercice 1.** ***(recherche d'un élément dans un ensemble trié)*** *Écrire la fonction récursive* `DichoRecursive` *qui recherche par dichotomie un produit* `p` *dans un ensemble trié et renvoie l'indice de la case contenant* `p` *si* `p` *est présent dans l'ensemble,* -1 *sinon. Évaluer sa complexité.*

▸ **Exercice 2.** ***(suppression d'un élément dans un ensemble trié)*** *Écrire la fonction* `Suppression` *qui supprime un produit* `p` *d'un ensemble* `e` *trié passé en paramètre. La fonction renverra 1 si la suppression s'est bien passée, 0 sinon.*

▸ **Exercice 3.** ***(construction d'un ensemble trié)*** *Nous allons construire un tableau trié de n éléments en considérant que si l'on dispose d'un tableau trié de $n-1$ éléments, il suffit d'insérer le n^{ieme} élément à la bonne place dans ce tableau. Ainsi le tableau trié de n élément sera construit par insertions successives des n éléments.*

Écrire la fonction `AjoutTrie` *qui insère un produit* `p` *à sa bonne place dans un ensemble* `e` *trié passé en paramètre. La fonction renverra 1 si l'insertion s'est bien passée, 0 sinon. Évaluer la complexité de l'insertion et celle de n insertions successives.*

▸ **Exercice 4.** ***(union, intersection de deux ensembles triés)*** *Écrire les fonctions suivantes qui renverront 1 si l'opération s'est bien passée, 0 sinon :*

- `Union` *qui construit l'union triée de deux ensembles triés ;*
- `Intersection` *qui construit l'intersection triée de deux ensembles triés.*

▸ **Exercice 5.** ***(inclusion d'un ensemble trié dans un autre)*** *Écrire la fonction* `Inclusion(Ensemble e1, Ensemble e2)` *qui renverra -1 si* `e1` *est inclus dans* `e2`*, 1 si* `e2` *est inclus dans* `e1` *et 0 sinon.*