

Devoir surveillé – SD- licence IUP ASR

Aucun document autorisé

14/12/2004

Conventions communes à tous les exercices

Les tableaux seront représentés comme en cours/td par le type suivant:

```
typedef struct ttableau{
    int tailleMax;
    int taille;
    int * tab;
} tableau;
```

Les listes seront représentées comme en cours/td par le type suivant:

```
typedef struct telement{
    int valeur;
    struct tliste * suiv;
} element;
typedef noeud * liste;
```

Les arbres binaires seront représentés comme en cours/td par le type suivant:

```
typedef struct tnoeud{
    int valeur;
    struct tnoeud * gauche;
    struct tnoeud * droit;
} noeud;
typedef noeud * arbre;
```

Exercice 1: tableaux

Donnez un algorithme qui compte le nombre d'éléments strictement positifs d'un tableau. Quelle est la complexité de votre algorithme ?

Exercice 2: Arbres binaires de recherche

Question 1

Donnez la définition d'un arbre binaire de recherche.

Question 2

Donnez l'algorithme déterminant le plus grand élément d'un arbre binaire de recherche. Quelle est la complexité de votre algorithme ?

Question 3

Donnez l'algorithme permettant de supprimer le plus grand élément d'un arbre binaire de recherche.

Exercice 3: doublons

Question 1

Soient t1 et t2 deux tableaux triés et sans redondances. Ecrire un algorithme permettant d'obtenir un nouveau tableau contenant les éléments communs à t1 et à t2. Quelle est la complexité de votre algorithme ?

Question 2

Faire de même pour deux listes chaînées triées: Ecrire un algorithme permettant d'obtenir une liste contenant les éléments communs à t1 et t2.

Question 3

On suppose que nos listes chaînées ne sont pas triées. Ecrire un algorithme permettant d'obtenir une liste contenant les éléments communs à t1 et t2.