Algorithmique 1

DS 3

Arbres Binaires, Tas et Arbre Binaires de Recherches

Exercice 3.1

1. Construire un tas (max) avec les valeurs successives suivantes :
   22 12 6 3 1 4 8 16 10 25 5
   On détaillera chaque étape de construction.
2. Faire la trace de l'exécution de la fonction supprimerValeur appliquée au tas construit ci-dessus,

Exercice 3.2

Modifier l'algorithme d'ajout dans un arbre binaire de recherche pour ne pas rajouter de doublons (sommet ayant la même information) sans test d'appartenance préalable

Exercice 3.3

On suppose donner un arbre binaire représentant une expression arithmétique contenant des entiers et des opérateurs binaires + et * . Si l'expression est bien formée (correcte) alors l'arbre binaire est complet. Chaque feuille de cet arbre contient donc un entier et chaque noeud interne un opérateur (l'un des symboles + ou *). On adopte comme convention que le symbole + est codé par -1 et * par -2..

Ecrire une fonction `valeurExpression( val A: arbreBinaire)` qui évalue la valeur de l'expression représentée par un arbre binaire si l'arbre est complet sinon renvoie `NIL`, en visitant une et une seule fois tous les noeuds de l'arbre (un seul parcours de l'arbre).