

Tris et arbres binaires

Exercice 1 : Arbre binaire de recherche Pour trier de façon croissante une suite d'éléments, on se propose d'effectuer un tri par insertion en utilisant un arbre binaire de recherche. On rappelle qu'un arbre binaire de recherche est caractérisé par les propriétés suivantes :

- tous les noeuds du sous-arbre *gauche* d'un noeud donné ont une valeur *inférieure ou égale* à celle du noeud donné ;
- tous les noeuds du sous-arbre *droit* d'un noeud donné ont une valeur *supérieure* à celle du noeud donné.

La recherche d'une valeur dans l'arbre se déroule de la façon suivante (l'insertion d'une nouvelle valeur suit un schéma similaire) :

1. on teste d'abord si le contenu de la racine est égal à la valeur recherchée ;
2. si ce n'est pas le cas, on recommence récursivement la recherche dans l'arbre de gauche si la valeur est plus petite que le contenu de la racine, ou dans l'arbre de droite dans le cas contraire.

Compléter le fichier `~scheuer/LicMath/Sujets/insert.p` pour que le programme correspondant effectue le tri prévu.

Exercice 2 : Tri par tas On veut trier par ordre croissant les éléments d'un tableau en utilisant un tri par tas. On rappelle qu'un tas est un tableau représentant un arbre ayant la propriété que la valeur de chaque noeud est supérieure ou égale à celle de ses fils.

À l'aide du fichier `~scheuer/LicMath/Sujets/tas.p`, définir un programme effectuant un tri par tas.