



TD N°3
Les Structures de Données

Exercice 1

1. Dessiner l'arbre binaire de recherche obtenu en insérant les éléments de la liste suivante dans leur ordre d'arrivée : 30, 40, 23, 58, 48, 26, 11, 13, 20.
2. Donner le résultat des parcours préfixe, infixé et postfixe de l'arbre précédent.
3. Donner l'arbre précédent après la suppression des clés suivantes : 48 puis 23.
4. Ecrire la fonction **NB_Sup (R: pointeur(Noeud), x : entier)** : entier qui retourne le nombre de clés supérieures à x dans l'arbre binaire de recherche de racine R.
5. Ecrire la fonction **Hauteur (R: pointeur(Noeud))** : entier qui retourne la hauteur de l'arbre binaire de recherche de racine R.
6. Ecrire la fonction **Equilibré(R : Pointeur(Noeud))** : Boolean qui retourne vrai si l'arbre de recherche binaire de racine R est équilibré et faux sinon.

Exercice 2

On souhaite construire un arbre AVL en insérant successivement les valeurs suivantes :

45, 20, 10, 25, 30, 50, 60

1. Dessiner l'arbre après chaque insertion.
2. Indiquer, après chaque opération, si une rotation est nécessaire (gauche, droite ou double) et la réaliser.
3. À la fin, donner l'arbre final équilibré.

Exercice 3

1. Étant donné une liste d'entiers de taille n, comment peut-on la trier (ordre décroissant) en utilisant un TAS ?
2. Appliquer le tri par TAS sur la liste suivante (illustrer toutes les permutations possibles) : L = {34, 20, 40, 10, 7, 15, 22, 8, 4}
3. Calculer la complexité au pire des cas du tri par TAS.

Exercice 4

1. Écrire la version itérative du tri rapide, à l'aide d'une pile.