

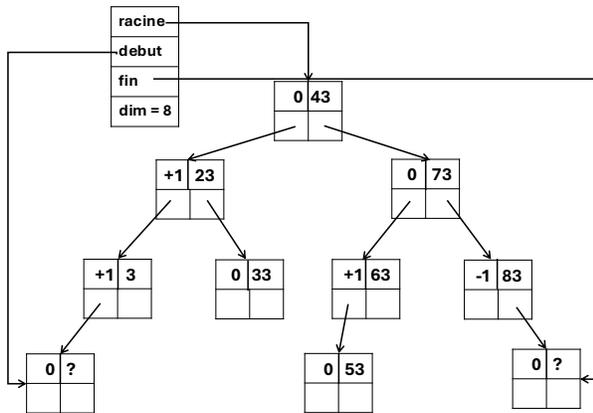
UNIVERSITÉ DE SHERBROOKE
DÉPARTEMENT D'INFORMATIQUE

IFT 339

Série d'exercices - Thème #10 : Équilibre et Itération dans les arbres binaires de recherche

Exercice 1 :

Considérer la représentation ci-après d'un arbre AVL. Cette représentation ne comporte pas de membre **parent** dans un noeud.



La figure présente un cas général un cas général contenant 8 éléments : 3, 23, 33, 43, 53, 63, 73, et 83.

```
template <typename TYPE>
class noeud{
private:
    TYPE val;
    int indice// h_gauche - h_droite
    noeud* gauche;
    noeud* droit;
public:
    noeud(TYPE&);
    noeud(TYPE&, noeud*, noeud*);}
```

```

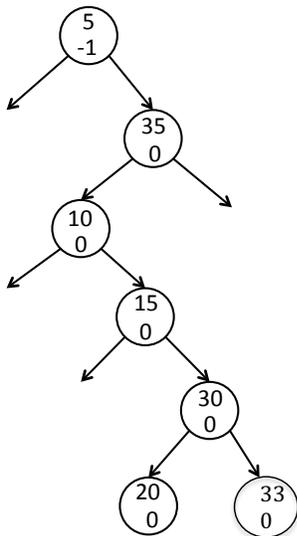
template <typename TYPE>
class ABR{
private:
    noeud *racine, debut, fin;
    size_t dim;
public:
    ...
    int erase(noeud*, TYPE x)//retourne un indice de noeud
}

```

Écrire l'algorithme de la fonction récursive **erase** qui permet de retirer un élément x se trouvant dans une feuille de l'arbre AVL. La complexité en temps dans le pire des cas doit être $O(\log(n))$.

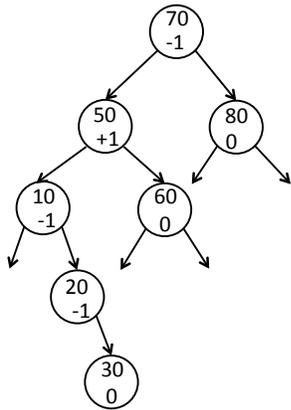
Exercice 2

Dessiner les étapes de l'insertion de 18 puis 34 dans cet arbre AVL.



Exercice 3 :

Dessiner les étapes de la suppression de 50 de cet arbre AVL.



Exercice 4 :

Quel est le nombre minimum d'éléments possible dans l'arbre AVL de l'exercice 3 ? Quel est le nombre maximum d'éléments possible ?