

1 Arbres binaires de recherche

Un arbre binaire de recherche (ABR) est un arbre pour lequel chaque nœud vérifie les deux propriétés suivantes :

1. chacun des nœuds de son sous arbre gauche a une étiquette de valeur inférieure (ou égale) à lui même,
2. chacun des nœuds de son sous arbre droit a une étiquette de valeur supérieure à lui même

Dans la suite du TD on utilisera les classes suivantes :

```
class Arbre {
    private Noeud racine;
    public Arbre () { this.racine = null; }
    public Arbre (Noeud n) {
        this.racine = n;
    }
}

class Noeud {
    private int etiquette;
    private Noeud gauche;
    private Noeud droit;
    public Noeud(int e, Noeud g, Noeud d) {
        this.etiquette = e;
        this.gauche = g;
        this.droit = d;
    }
}
```

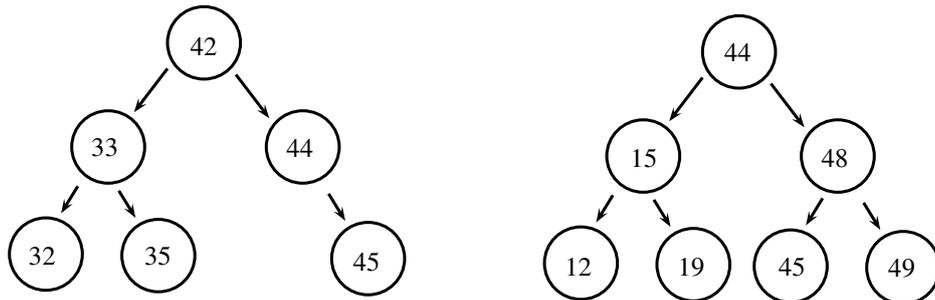
Écrire les méthodes suivantes dans la classe `Arbre` :

1. Écrire une méthode `public boolean verifier()` qui s'assure que l'arbre `this` est un ABR.

Dans les questions suivantes on suppose (et il faudra le vérifier) que l'arbre est un ABR.

2. Proposer une méthode d'affichage `public void afficher()` qui affiche les étiquettes des nœuds de `this` de telle sorte qu'ils apparaissent dans l'ordre croissant.
3. Proposer une méthode `public int maximum()` retournant l'étiquette de valeur maximale de l'ABR. Même question avec le minimum.
4. En utilisant la structure de l'ABR, proposer une méthode de recherche d'un entier dans `this`. La méthode renverra un booléen indiquant la présence dans l'arbre de l'entier `n` donné en paramètre.
5. On veut à présent insérer un élément dans l'arbre tout en s'assurant que l'arbre obtenu soit encore un ABR. Proposez une méthode `public void ajouter(int n)` qui ajoute un nœud d'étiquette `n`.
6. Dessiner l'arbre obtenu si on insère un à un les nœuds 10, 5, 8, 3, 13, 2, 1.

7. Expliquer comment supprimer le nœud d'étiquette 44 sur les deux arbres suivants :



8. Montrer que l'algorithme suivant permet de supprimer un nœud tout en conservant la structure de l'ABR :

- Si le nœud à supprimer ne possède pas de fils, on l'enlève
- Si le nœud à supprimer n'a qu'un fils, on remplace le nœud par son fils
- Si le nœud à supprimer possède deux fils, on le remplace par le sommet de plus petite valeur dans le sous arbre droit, puis on supprime ce sommet

Implémenter cet algorithme.

9. Écrire une fonction `public int[] tri (int[] t)` qui retourne un nouveau tableau correspondant au tableau `t` trié en utilisant un ABR.