

# Exercice 13

---

## ListeSort

Obectif : Faire le développement de la classe ListeSort qui est une variante d'implémentation d'une liste ordonné sous la forme d'un arbre binaire infixé

### 1. CAHIER DES CHARGES 2

## 1. Cahier des charges

Nous voulons implémenter une liste d'entier (classe **ListeSort**) dont les éléments ne sont pas stockés dans un tableau mais dans un un arbre binaire.

La méthode de la classe ListeSort qui ajoute un nouvel élément dans la liste, utilise l'arbre, suivant le principe suivant :

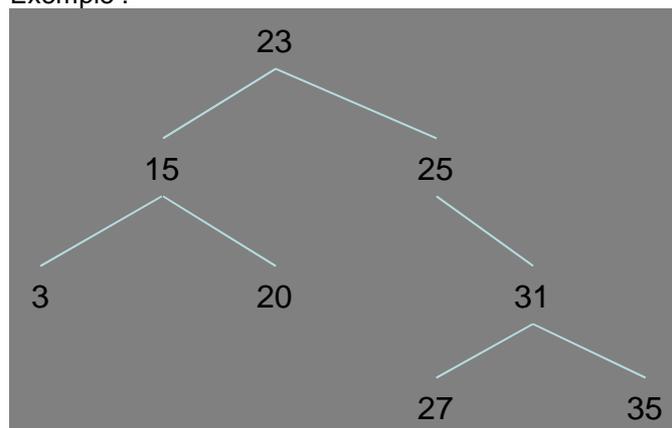
- la racine de l'arbre est le premier élément ajouté dans la liste.
- pour ajouter un nouvel élément, on parcourt récursivement l'arbre en testant si la valeur de l'élément est inférieure à celui du nœud courant. Si inférieur alors on part à gauche sinon on part à droite. On ajoute l'élément sous le dernier nœud rencontré à gauche ou à droite (si inférieur ou non du nœud).

On s'aperçoit que si on parcourt un tel arbre suivant un parcours infixe, les éléments sont ordonnés croissants.

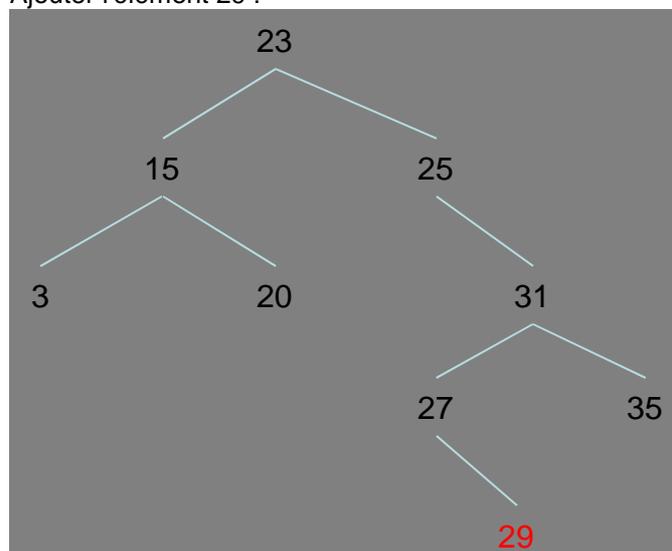
Rappel : parcours infixe = on parcourt à gauche, puis on affiche le nœud, puis on parcourt à droite.

De plus cela permet de rechercher un élément qui existe ou qui n'existe pas très rapidement. Par exemple, sur un arbre équilibré de 1023 éléments (profondeur de 11), il faut au plus 11 tests pour trouver le nombre.

Exemple :



Ajouter l'élément 29 :



Créer la classe ListeSort avec les méthodes suivantes :

- le constructeur (sans paramètre) qui crée une liste vide (l'arbre = null)
- la méthode void add(int v) qui ajoute un nouvel élément
- la méthode void afficher qui affiche les éléments dans l'ordre infixé (ou croissant)
- la méthode boolean search(int v) qui recherche un élément (retourne true ou false)
- la méthode Iterator<int> iterator qui retourne un itérateur permettant de parcourir les éléments dans l'ordre croissant (next hasNext)

Voir :

[http://fr.wikipedia.org/wiki/Arbre\\_binaire\\_de\\_recherche](http://fr.wikipedia.org/wiki/Arbre_binaire_de_recherche)