

Nous rappelons que les méthodes sont maintenant à écrire dans la classe correspondante.

Sauf mention contraire, il est interdit d'utiliser des méthodes des exercices précédents pour résoudre l'exercice courant. Les boucles sont interdites.

Conseil pour la conception de votre algorithme :

- la méthode du cours indique qu'il faut d'abord penser à une "grande entrée" : ici, un "gros" arbre sera un arbre dont les deux sous arbres sont non vides
- ensuite, vérifiez que votre code précédent est toujours correct lorsque
 - l'arbre est vide (ce qui n'est en général pas le cas, et entraîne donc l'ajout d'un cas de base)
 - l'arbre est non vide, mais l'un (ou les deux) sous arbres sont vides (ce qui peut entraîner soit des ajouts de cas de base, soit d'autres branches qui feront des appels récursifs différemment)

1 Définition

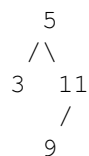
Un arbre A est un arbre binaire de recherche (ABR) ssi

- A est vide
- ou
- $A.\text{filsG}$ est un ABR, et
 - $A.\text{filsD}$ est un ABR, et
 - toute valeur x dans $A.\text{filsG}$ vérifie $x \leq A.\text{val}$, et
 - toute valeur x dans $A.\text{filsD}$ vérifie $x > A.\text{val}$

Exercice 1. Echauffement

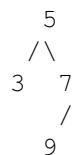
Question 1.1.

Est ce que l'arbre ci-dessous est un ABR ?



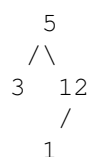
Question 1.2.

Est ce que l'arbre ci-dessous est un ABR ?



Question 1.3.

Est ce que l'arbre ci-dessous est un ABR ?



Exercice 2. Recherche

Question 2.1.

Ecrire la méthode `boolean recherche(int x)` suivante. Prérequis : l'arbre courant est un ABR.

Action : retourne vrai ssi x est dans l'arbre.

Exercice 3. Affichage trié

Question 3.1.

Ecrire la méthode `String toStringTrie()` suivante. Prérequis : l'arbre courant est un ABR.

Retourne dans une chaîne toutes les valeurs de l'arbre triées par ordre croissant.

Exercice 4. Insertion

Question 4.1.

Ecrire la méthode `void insert(int x)` suivante. Prérequis : l'arbre courant est un ABR. Action :

insert x dans l'arbre de telle sorte que l'arbre reste un ABR.

Exercice 5. Suppression

Question 5.1.

Ecrire la méthode `void suppr(int x)` suivante. Prérequis : l'arbre courant est un ABR. Action : si x est présent, le supprime de l'arbre de telle sorte que l'arbre reste un ABR. Vous avez le droit d'écrire des méthodes auxiliaires de votre choix.