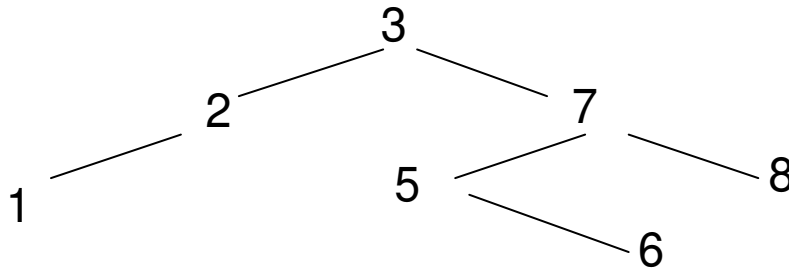


Structure de données avancées TP -Arbres binaires de Recherche

Manipulation de type générique et de méthodes récursives pour les Arbres Binaires de Recherche

Un arbre binaire de recherche est un arbre binaire où les éléments stockés peuvent être comparés entre eux et où :

- tout élément contenu dans le fils gauche est inférieur ou égal à l'élément de l'arbre
- tout élément contenu dans le fils droit est supérieur ou égal à l'élément de l'arbre



1. Reprenez la classe `ArbreBinaire` de la séance précédente et implémentez la méthode `static estABR (AB<F> ab)` prenant en argument un arbre binaire d'éléments comparable entre eux, et retournant vrai si `ab` est un arbre binaire de recherche. Réfléchissez tout d'abord à sa signature
Test : vérifier la méthode sur un arbre binaire d'entiers et un arbre binaire de disques préférés.
2. Implémentez la classe arbre binaire de recherche `ABR<E comparable<E>>` ne prenant en paramètre que des objets comparables entre eux.
Test : Essayer de créer un ABR de disque, créer un ABR de Disques préférés
3. Implémentez une méthode permettant de rechercher si un élément est contenu dans un `ABR<E>`.
Test : vérifier la méthode de recherche sur un ABR d'entiers et un ABR de disques préférés.
4. Implémentez une méthode permettant d'ajouter un élément à un `ABR<E>`
Test : vérifier la méthode d'ajout sur un ABR d'entiers et un ABR de disques préférés.
5. Implémentez une méthode permettant de supprimer un élément dans un ABR (on supposera que chaque valeur apparaît au plus une fois)
Test : vérifier la méthode de suppression sur un ABR d'entiers et un ABR de disques préférés.
6. Reprenez les méthodes et partout où vous avez utilisé `e1.equals(e2)` changez par `e1.compareTo(e2)` Qu'est-ce que ça change ? Pour les Arbres d'entier, de Disques Préférés