

Les arbres III : arbres binaires de recherche

Capacités attendues

- ✓ Rechercher une clé dans un arbre de recherche, insérer une clé ;
- ✓ La recherche dans un arbre de recherche équilibré est de coût logarithmique.

Définition

Un arbre binaire de recherche (ABR) est un arbre binaire dans lequel les étiquettes, appelées **clés**, peuvent être comparées entre elles, et telles que, pour chacun des nœuds :

- sa clé est supérieure ou égale à toutes les clés du sous-arbre gauche ;
- sa clé est inférieure ou égale à toutes les clés du sous-arbre droit.

Exercice 1

Représenter un arbre binaire satisfaisant les conditions ci-dessus pour être un ABR.

Exercice 2

Construire un ABR à partir des nombres suivants, en les insérant les uns à la suite des autres : 25, 11, 31, 4, 9, 42, 30, 28, 18.

1 Algorithmes

Dans un ABR, les nœuds sont ordonnés :

- ceci facilite la recherche d'une clé dans l'arbre ;
- l'insertion d'un nouvel élément ne peut pas se faire au hasard.

, ce qui facilite les recherches.

Recherche dans un ABR

On parcourt l'arbre en commençant par la racine.

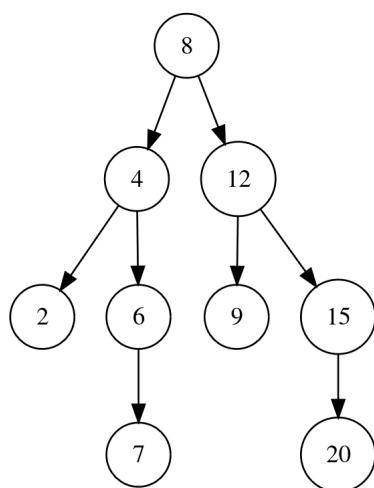
- si elle est égale à l'élément cherché, **la valeur est trouvée** ;
- si l'élément cherché est plus petit, **on parcourt le sous-arbre gauche** ;
- si l'élément cherché est plus grand, **on parcourt le sous-arbre droit**.

Insertion dans un ABR

On applique le même algorithme que pour rechercher un élément, sauf que lorsqu'on arrive sur un sous-arbre vide, on le remplace par un sous-arbre ayant la valeur à insérer comme racine.

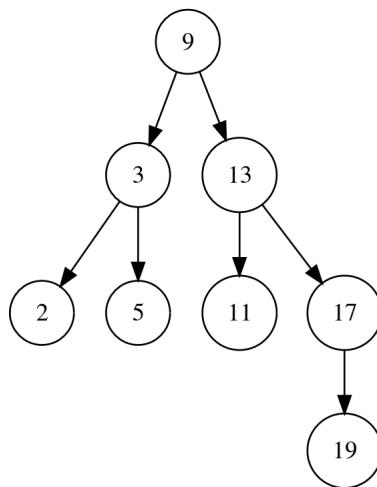
Exercice 3

En appliquant ces algorithmes, rechercher 6 et 3 puis insérer 10 et 1 dans l'ABR suivant.



Exercice 4

On considère l'arbre suivant.



- Justifier qu'il s'agit bien d'un arbre binaire de recherche.

- Appliquer l'algorithme d'insertion pour y ajouter le chiffre 7.

- Appliquer l'algorithme de recherche pour trouver les nombres 11 et 14.

- Donner les parcours en profondeur préfixe, infixé et suffixé de cet arbre. Quel parcours permet d'obtenir la liste des nombres dans l'ordre croissant ?

- Donner le parcours en largeur de cet arbre.

Exercice 5

Dans tout l'exercice, on se basera sur l'ABR défini dans l'exercice 4 ainsi que sur les résultats de recherche et d'insertion obtenus dans les questions.

1. Récupérer sur notre site le fichier `abr.py` et le mettre dans le même dossier que `noeud.py` et `displayer.py`. Commenter toute la partie `# SCRIPT` de `noeud.py`
2. Ajouter l'import nécessaire au fonctionnement.
3. Dans la partie `# SCRIPT`, continuer la construction de l'ABR pour qu'il corresponde à celui de l'exercice 4.
4. Compléter la méthode `rechercher(self, cle)` dans la classe ABR. Elle renvoie `True` si l'élément a été trouvé, `False` sinon.
5. Compléter la méthode `inserer(self, cle)` dans la classe ABR. Elle ne renvoie rien de spécial.

Exercice 6 – (Questions subsidiaires)

Comment pourrait-on représenter les arbres en Python sans POO ?

Noter quelques exemples.