

Table de hachage

CPE

4ICS

1 But

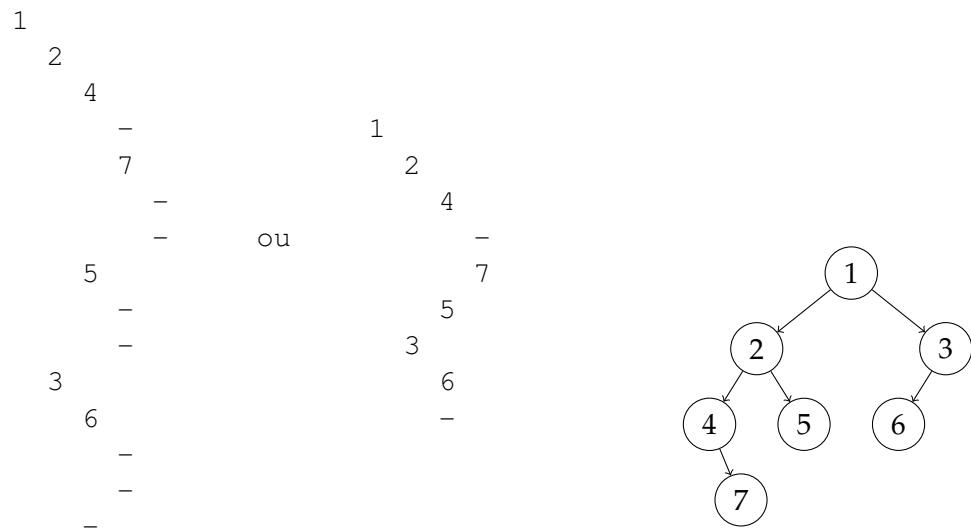
L'objectif est de développer en python un module offrant le type abstrait **ABR**. L'objet devra disposer des méthodes :

- constructeur (`__init__(self)`)
- `insert(value)` - en place, ne fait rien si value existe déjà
- `delete(value)` - en place
- `exist(value)` - renvoie vrai ou faux
- `isEmpty()` - renvoie vrai ou faux
- `clear()` - en place
- `union(autre_abr)` - renvoie un arbre
- `intersection(autre_abr)` - renvoie un arbre
- `displayInfixe()` -
- `displayPrefixe()` -
- `displayBFS()` - parcours en largeur
- `heightTree()` - renvoie la hauteur de l'arbre

Dans le cas où vous avez besoin d'une méthode intermédiaire, faites préfixer un double underscore devant le nom de la fonction.

Essayez de faire la moitié des méthodes en récursif et l'autre moitié en itératif.

Pour la méthode d'affichage prefixe, essayez d'obtenir ce type d'affichage :



2 Performances

Tester plusieurs fois l'ajout de 500 entiers aléatoires entre 0 et 10000. Quelles sont les hauteurs d'arbre obtenus? Tester votre code pour obtenir le profil du temps de recherche de 100 valeurs (entre 0 et 10000) en fonction de la hauteur de l'arbre. Combien de nœuds sont visités en moyenne? Que pouvez vous en dire?