Partie 1 – classe AB

Vous allez coder la classe AB avec les attributs suivants :

- racine qui est une liste (pour le moment avec un seul élément dans la liste)
- gauche qui est un AB
- droite qui est un AB

(nous voilà dans le récursif…)

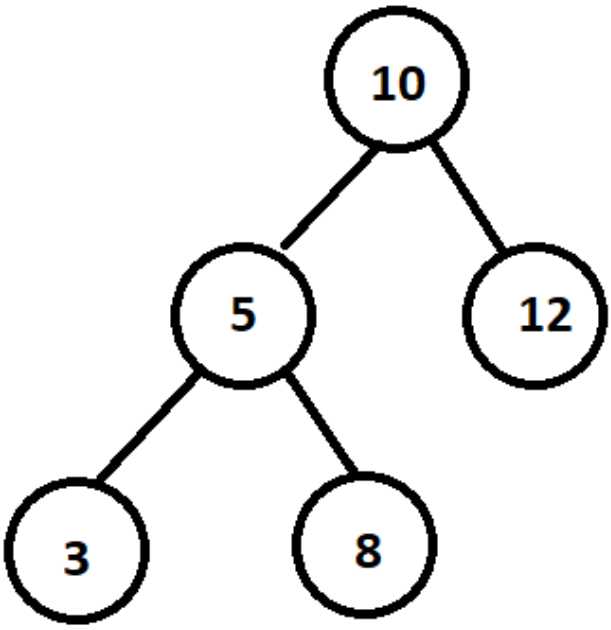
1. Ecrire le code du constructeur qui prend 3 paramètres (optionnels), les setters (en particulier celui de la racine)

On notera qu'un arbre vide possède comme racine une liste None (et non une liste vide) et cela, pour nous faciliter le codage…

2. Ecrire le code de la méthode estVide() qui nous retourne un booléan pour nous dire si l'arbre est vide ou pas.

Dans votre fichier main.py qui contiendra le corps de vos appels :

3. Créer A1 qui sera l'arbre vide et testez votre méthode estVide()
4. Créer A2 qui possèdera juste une racine avec la valeur 5 et testez votre méthode estVide()
5. Créer A3 qui possèdera juste une racine avec la valeur 3
6. Modifier la partie gauche de A2 pour y placer A3
7. Créer l'arbre Atest qui ressemble à celui-ci, et testez votre méthode estVide()



8. Rajoutez ensuite dans votre classe la méthode taille() qui retourne un entier représentant la taille de l'arbre
9. Testez cette méthode sur Atest
10. Cette méthode est elle récursive ? Si non, fonctionne t elle ? Il est temps de la coder en récursif
11. Codez la méthode prefixe() qui affiche le parcours préfixe d'un AB
12. Testez cette méthode
13. Codez les 2 autres parcours et testez les.
14. Codez la méthode hauteur() qui nous retourne la hauteur de l'arbre
15. Codez la méthode estABR() qui retourne un booleen
16. Codez la méthode estEquilibre() qui retourne un booleen