

# TP ABR

1. Proposez une structure de données Python Objet, pour représenter un ABR.
2. Implémentez un exemple numérique d'ABR contenant une dizaine de valeurs numériques et définissez la liste contenant les mêmes valeurs.
3. Implémentez la méthode « existe » vue en TD ainsi que sa version équivalente sur une liste (qui recherche si un élément existe dans une liste). Pour les deux versions, vous afficherez un compteur de comparaisons qui compte le nombre de fois que l'élément cherché est comparé à un élément de la structure.
4. Expérimentez la recherche, avec les deux versions, pour chaque valeur de la structure et donnez la moyenne de comparaison pour chaque version. Quelle serait cette moyenne, pour chaque version, si vous aviez N valeurs ?
5. Implémentez le constructeur d'ABR, prenant en paramètre une liste de valeurs numériques.
6. Implémentez la méthode d'ABR « add\_node » qui prend en paramètre une étiquette et ajoute dans l'ABR (à la bonne place) un nouveau nœud qui contient l'étiquette.
7. Implémentez la méthode d'ABR « balance » qui retourne un nouvel ABR équilibré.
8. Généralisez votre code pour que votre ABR soit capable de manipuler une liste d'étiquettes quelconques. Les étiquettes que vous manipulerez devront cependant être comparable, c'est à dire implanter les méthodes « `__eq__(self, other)` » et « `__lt__(self, other)` » qui retournent respectivement « True » si « self » est égale à « other » et « True » si « self » est plus petit que « other ».