

TD d'Algorithmique et structures de données

Arbres Binaires de Recherche

Équipe pédagogique Algo SD

Soit A un arbre binaire. Il est représenté par un objet comportant un champ VALEUR et un tableau de deux FILS, ce qui s'écrit en Python :

```
class Noeud:
    def __init__(self, valeur, fils_gauche=None, fils_droit=None):
        self.valeur = valeur
        self.fils = [fils_gauche, fils_droit]
```

1 Vérification qu'un arbre binaire est un ABR

1. Ecrire en Python ou en pseudo-code la fonction récursive $VERIF(self)$ qui vérifie si un arbre binaire est un ABR, et dans ce cas renvoie les éléments minimum et maximum.

2 Suppression dans un ABR

On s'intéresse à la suppression d'un élément dans un ABR. Comme pour l'insertion, elle nécessite de mettre à jour la structure de l'arbre. Les modifications doivent être réalisées de manière à conserver les invariants de structure et d'ordre.

1. Commencez par essayer quelques exemples de suppression manuellement. Y-a-t'il des cas "faciles" ?
2. Écrire en Python ou en pseudo-code un algorithme $DELETE(self, key)$ qui supprime un élément key dans un ABR.
3. Donner le coût (en meilleur cas, en pire cas) de la primitive $DELETE$ en fonction du nombre d'éléments n dans l'arbre.

3 Sous-ensemble d'un ABR

1. Écrire en python ou en pseudo-code un algorithme $ITERATE(Min, Max, A)$ qui itère sur les éléments de A entre Min et Max .