

1. Introduction

- Dans cette feuille d'exercices, on s'intéresse aux opérations sur les arbres binaires de recherche (ABR)
 - Recherche, insertion et suppression
 - Rotations gauche et droite pour équilibrer les ABR/

2. CRUD sur les ABR

Dans cet exercice, on implémente les algorithmes CRUD pour les ABR.

2.1 Rechercher un élément

Ecrire la fonction ou la procédure permettant de vérifier si un élément appartient à un ABR.

2.2 Insérer un élément

Ecrire la fonction ou la procédure permettant d'insérer (s'il n'existe pas) un élément dans un ABR.

2.3 Supprimer un élément

Ecrire les fonctions ou procédures pour supprimer un élément dans un ABR. On utilisera le plus grand prédécesseur de l'élément à supprimer.

Annexe

Le type `ArbreKAire`

Méthodes de construction et d'initialisation

- fonction `arbreVide()` : `ArbreKAire`
- Fonction `creerArbre(Element, Entier)` : `ArbreKAire`

Méthodes de mise à jour

- procédure `modifierRacine (ArbreKAire (E/S), Element)`
- procédure `affEnfant (ArbreKAire (E/S), Entier, ArbreKAire)`
- procédure `supprimerFeuille(ArbreKAire (E/S), Entier)`

Méthodes d'accès

- fonction `estVide (ArbreKAire)` : `Booleen`
- fonction `recRacine(ArbreKAire)` : `Element`
- fonction `existeParent(ArbreKAire)` : `Booleen`
- fonction `recEnfant(ArbreKAire,Entier)` : `ArbreKAire`
- fonction `recParent(ArbreKAire)` : `ArbreKAire`
- fonction `recArite(ArbreKAire)` : `Entier`