

CHAPITRE 4

LES ARBRES BINAIRES DE RECHERCHE

4.1 Introduction

Un arbre binaire est une structure qui permet de regrouper un ensemble de données.

Un arbre binaire permet aussi de structurer les données de manière hiérarchique.

Si l'on veut localiser un élément parmi un grand nombre d'éléments dans un arbre binaire, y-a-t'il une manière de la faire de manière efficace ??

Peut-on organiser les données d'un arbre binaire de manière analogue à un tableau trié ??

4.2 Propriété fondamentale

Définition (Arbre Binaire de Recherche)

Un Arbre Binaire de Recherche¹ est une structure fondamentale pour la représentation d'ensembles dotés d'un ordre.

Un ABR est un AB particulier qui ordonne totalement les informations qu'il stocke :

- Toute les valeurs inférieures ou égales à celle de la racine sont stockées dans le descendant gauche de la racine.

1. noté également ABR

4.2 PROPRIÉTÉ FONDAMENTALE

- Toute les valeurs supérieures ou égales à celle de la racine sont stockées dans le descendant droit.

Ainsi, tout les nœuds à gauche d'un nœud valant x portent des valeurs inférieures ou égales à x et tout ceux à droite portent des valeurs supérieures ou égales à x .

Autrement dit, pour tout nœud n (valant x) de l'arbre, on s'assure que :

$$\max_val(n.sous_arbre_gauche) \leq n.val \leq \min_val(n.sous_arbre_droit) \quad (4.1)$$

l'inéquation 4.1 exprime la condition est appelée propriété fondamentale des ABR.

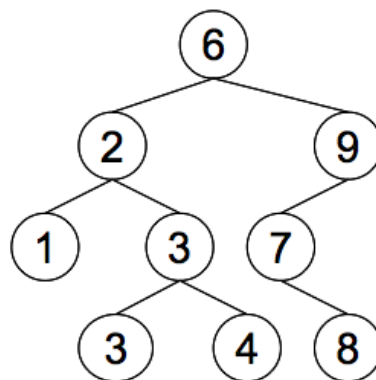


FIGURE 4.1 – Exemple d'un ABR

Remarque importante

Toute insertion ou suppression d'un nœud doit maintenir la propriété 4.1 vraie !

Motivation

Atout algorithmique : recherche et tri efficaces :

- tri car les données sont maintenues ordonnées ;
- recherche efficace, inspirée la dichotomie.

Exemple 1 ()

Le parcours infixe de l'arbre donne la liste triée.

Trier 3, 1, 5, 4, 7, 2.

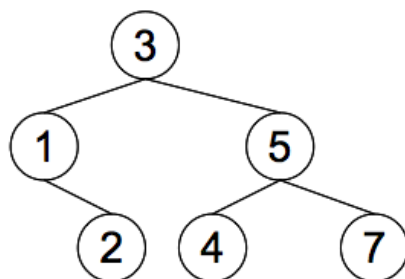


FIGURE 4.2 – Usage d'un ABR pour effectuer un tri

Remarque

Il existe une définition analogue d' ABR où on suppose que toutes les valeurs des nœuds sont distinctes.

Application 1 ()

1. Soit T un ABR vide.
Faire un schéma représentant T après l'insertion des nombres suivants : 10, 5, 12, 4, 11 (dans l'ordre).
2. Donner les listes des valeurs obtenus avec les parcours préfixé, postfixé et infixé.

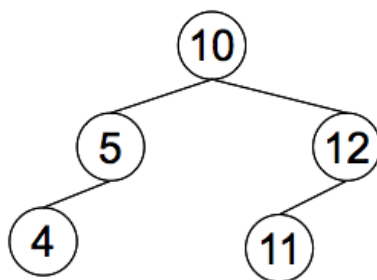


FIGURE 4.3 – Usage d'un ABR pour effectuer un tri

4.3 Recherche dans un ABR

Supposons qu'un ensemble soit représenté par un ABR.

La propriété fondamentale des ABR rend le test d'appartenance à l'ensemble particulièrement aisé.

Pour savoir si x est un élément de l'ensemble, on commence par comparer x à l'élément r situé à la racine de l'arbre :

- si $x = r$, la réponse à la question « x est-il dans l'ensemble ? » est immédiate et est positive ;
- si $x < r$, par la propriété fondamentale, x , à supposer qu'il soit dans l'ensemble, ne peut être qu'un descendant du fils gauche de la racine ;
- si $x > r$, x ne peut être qu'un descendant du fils droit de la racine.

La fonction `Recherche(x, A)` permettant de tester l'appartenance de l'élément x dans l'ABR A sera donc de nature récursif.

Application 2 ()

Énoncer la fonction `Recherche(x, A)`.

4.4 Insertion d'un élément dans un ABR

La procédure `INSERER(x, A)` permettant l'ajout de l'élément x à l'ensemble A peut être définie comme suit :

On commence par vérifier si A est vide :

- si c'est le cas, on crée un nouveau nœud pour contenir x et l'on fait pointer A dessus ;
- sinon, on cherche à positionner x un peu de la même façon que dans la fonction `Recherche()`.

Exemple 2 ()

Insertion de l'élément 6 dans l'arbre suivant :

4.5 Suppression d'un élément dans un ABR

Le principe de la procédure `SUPPRESSION(x, A)` permettant la suppression de l'élément x de l'ABR est le suivant :

- Si l'élément est une feuille, alors on supprime simplement ;

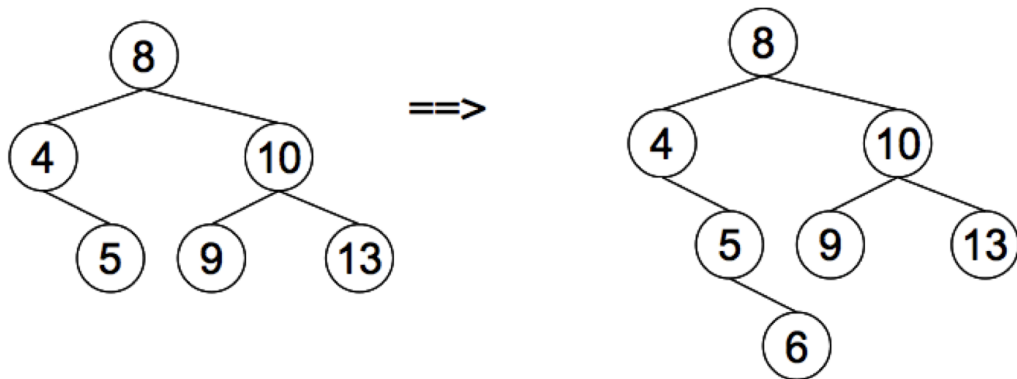


FIGURE 4.4 – Insertion dans un ABR

- Si l'élément n'a qu'un seul fils, alors on le remplace par son fils ;
- Si l'élément a deux fils, on le remplace au choix :
 - a/ soit par l'élément le plus à droite du sous-arbre gauche ;
 - b/ soit par l'élément le plus à gauche du sous-arbre droit, afin de conserver la propriété d'ordre.

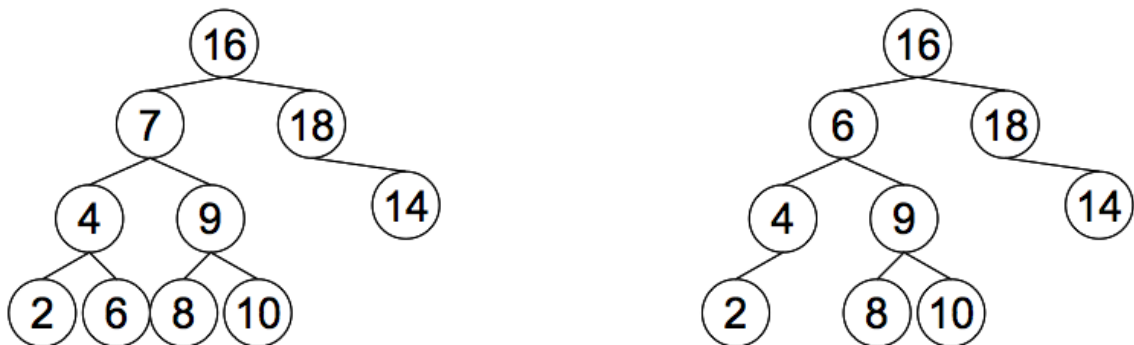


FIGURE 4.5 – Suppression d'un élément d'un ABR : cas 1

4.6 MISE EN ŒUVRE D'UN ABR

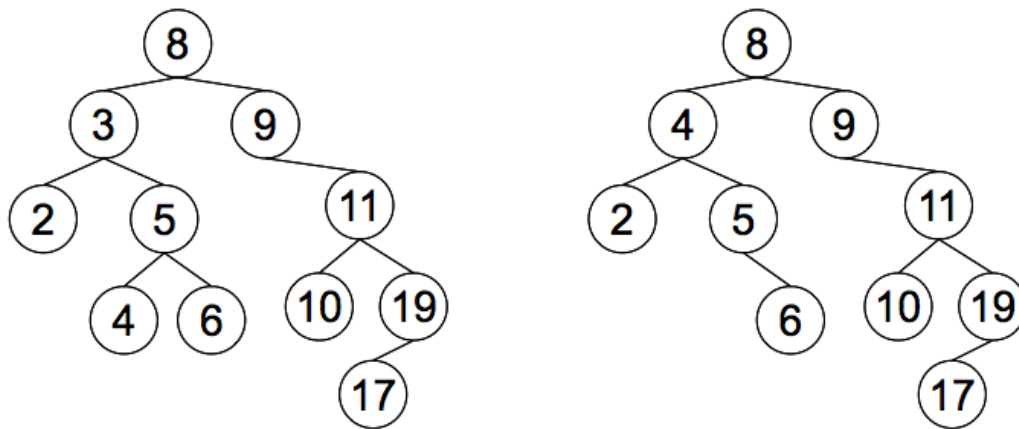


FIGURE 4.6 – Suppression d'un élément d'un ABR : cas 2

4.6 Mise en Œuvre d'un ABR

La mise en œuvre que nous avons définie pour les arbres binaires (voir page ??) est encore valable pour les ABR.

En effet, la différence entre les AB et les ABR n'est pas structurelle mais organisationnelle.