ALGORITHMES ET STRUCTURE DE DONNÉES
ARBRES BINAIRES
16/11/2022

Un arbre binaire de recherche est un arbre binaire étiqueté qui vérifie la propriété suivante :

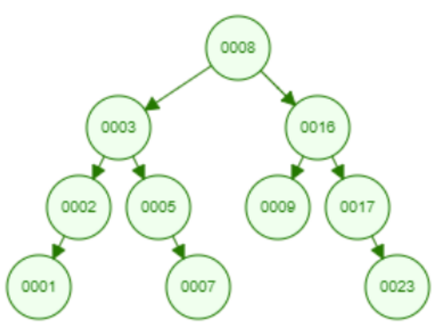
Pour tout nœud *v* de l'arbre...

- les étiquettes contenues dans le sous-arbre de gauche de v sont toutes strictement inférieures à l'étiquette de *v*
- les étiquettes contenues dans le sous-arbre de droitede v sont toutes strictement supérieures à l'étiquette de *v*

Remarque : Dans un ABR, les étiquettes sont toutes différentes.

Illustration :

Si on ajoute une à une les valeurs dans un ABR, on obtient l'arbre A. Les valeurs 8; 3; 16; 17; 23; 2; 5; 3; 1; 9; 7



Ajouter un élément :

Lorsqu'on souhaite ajouter un élément *t* dans un ABR, on part de la racine puis on "descend" de neoud en nœud jusqu'à trouver un emplacement vide.

A chaque nœud il faut choisir de continuer le chemin :

- à gauche si *t* < à l'étiquette du nœud;
- droite si *t* >
- on stoppe si *t* = l'étiquette du nœud ou si l'arbre est vide.

<u>Supprimer un élément</u> :

On le remplace par le nœud :

- qui est le plus petit nœud parmi ceux qui sont plus grands que lui dans l’ensemble,
- qui est le plus grand nœud parmi ceux qui sont plus petits que lui dans l’ensemble.

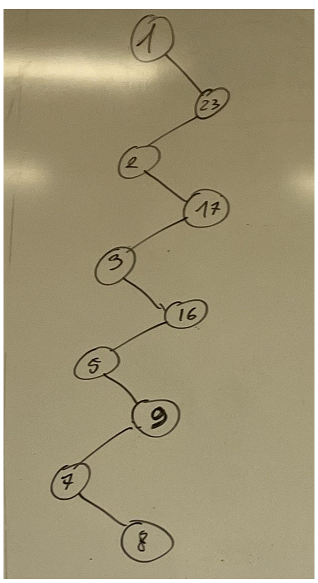
Problème :

Pour trouver où ajouter un nœud il faut déplacer jusqu’à :

- un emplacement vide,
- trouver la même valeur.

Ce qui peut prendre du temps si l’arbre possède une grande hauteur.

<u>Illustration :</u>



L’arbre suivant et l’arbre a représentent le même ensemble, cependant l’arbre ci-contre est moins intéressant car il possède une bien plus grande hauteur pour un même nombre de nœuds.

On souhaite au cours des ajouts et suppressions de nœuds conserver un arbre “équilibré”, il faut pour cela utiliser les arbres AVL.

Définition <u>arbre binaire équilibré</u> : on dit qu’un arbre binaire étiqueté est équilibré si :

- soit c’est un arbre vide,
- soit ses deux fils gauche et droit sont équilibrés et leur différence de hauteur ne dépasse pas 1.

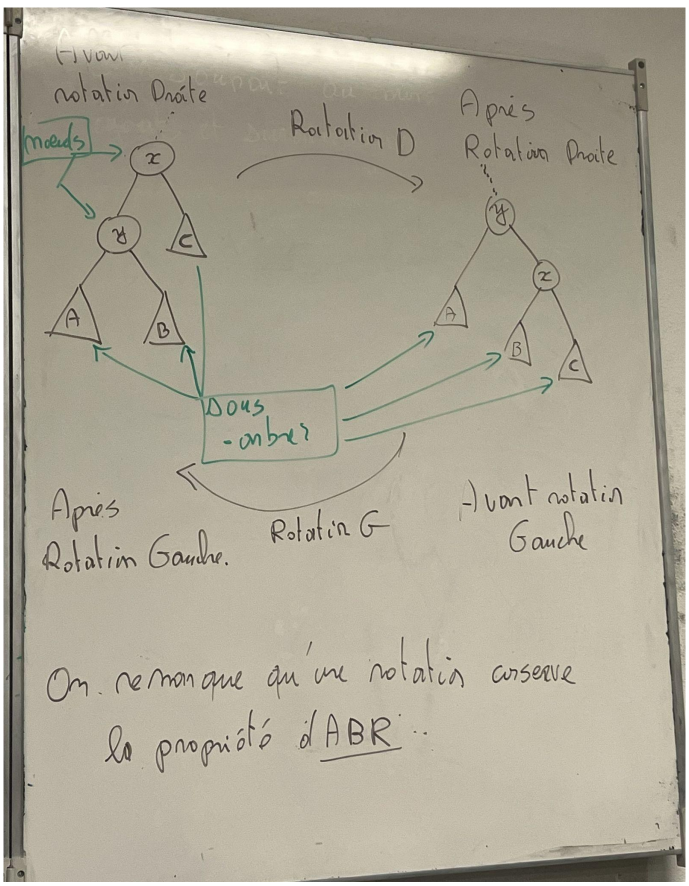
Théorème :

Tous les arbres binaires équilibrés de taille n (possédant n nœuds) ont une hauteur inférieure à **2 x log$_2$ (*n*)**

Le théorème a pour conséquence que tout ajout ou retrait dans un AVL de taille de *n* utilisent O (log *n*) opérations.

Problème :

Après un ajout classique ou retrait à un AVL la propriété d'équilibre peut être rompue. Il faut la rétablir par ce que l'on va appeler des rotations d'arbres.



Comment utiliser les rotations après un ajout ou suppression/extraction d'un nœud ?

L’ajout ou la suppression sont des opérations récursives. Nous allons voir le cas de la racine mais cela peut s’appliquer à n’importe quel sous-arbre.

Dans le cas d’un arbre AVL :

Soit l’arbre suivant après un ajout ou une suppression appliqué qu’à A

x

A B

hauteur a : hauteur du fils gauche

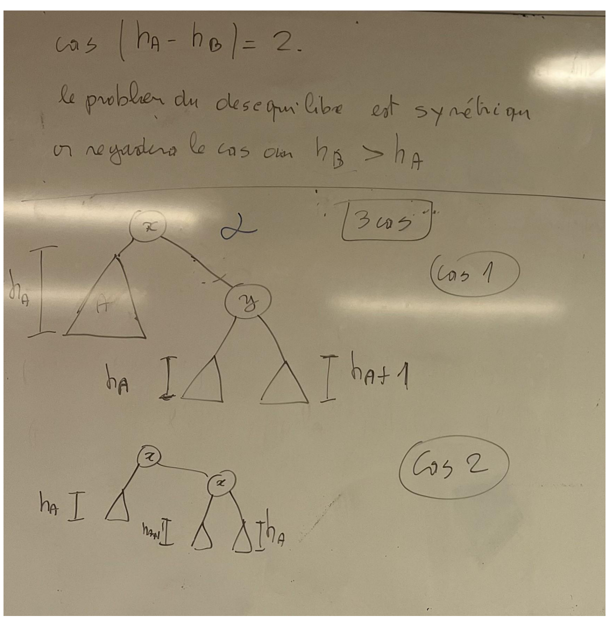
hauteur b : hauteur du fils droit

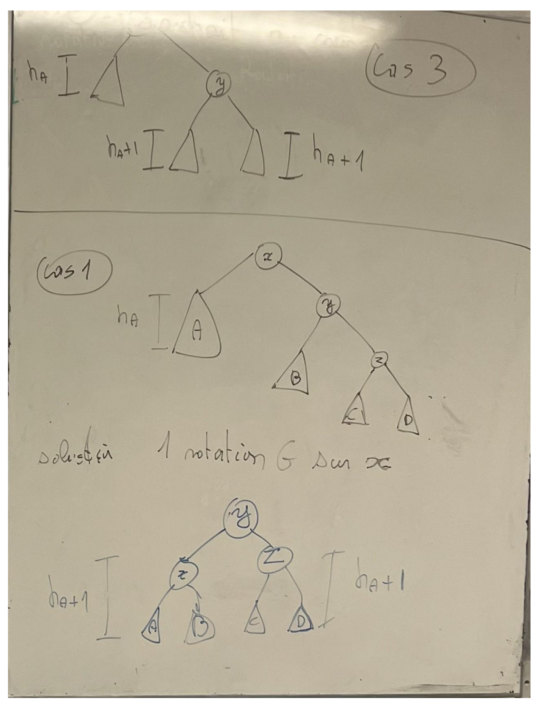
$|h_a - h_b|$ peut être égal. Possibilités : 0, 1, 2~~, 3, 4~~

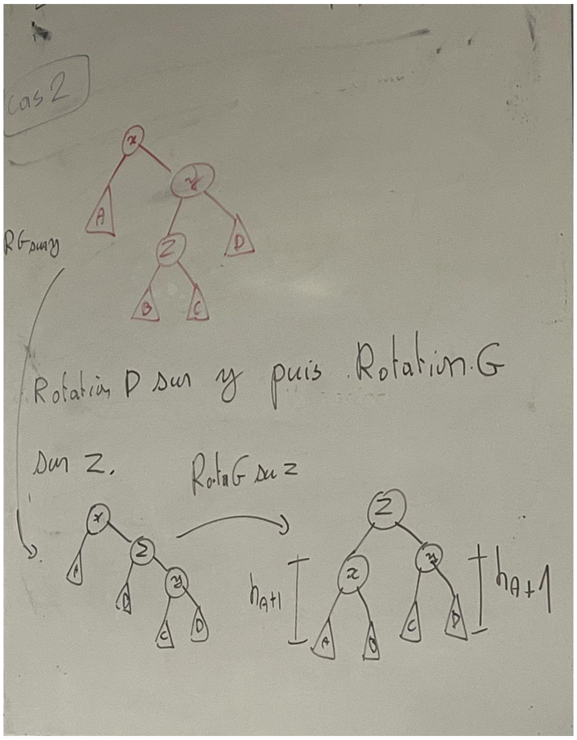
0 ou 1 ne pose pas de problème.

Pour $|h_a - h_b| = 2$.

Le problème du déséquilibre est symétrique. On regarda le cas où $h_b > h_a$







On mémorise dans chaque nœud *v* la hauteur du sous-arbre dont *v* est la racine.