AVL arbres

MPCI Spécialité informatique

Stéphane Grandcolas

Aix-Marseille Université

Contact : stephane.grandcolas@univ-amu.fr

Structure de données dictionnaire

Implémente les opérations :

- ajout,
- suppression,
- recherche.

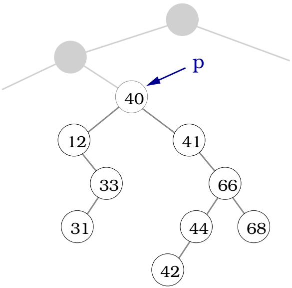
	ABR	AVL
ajout	$\mathcal{O}(h)$	$\mathcal{O}(\log n)$
suppression	$\mathcal{O}(h)$	$\mathcal{O}(\log n)$
recherche	$\mathcal{O}(h)$	$\mathcal{O}(\log n)$

h : hauteur de l'arbre, $\log n \leq h \leq n$ s'il y a n éléments.

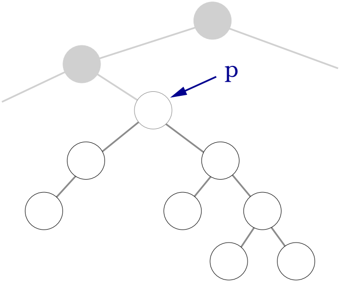
ABR (arbres binaires de recherche)

Définition. Un ABR est un arbre binaire qui satisfait l'ordre *infixe*, i.e. pour tout noeud $p = \langle x, G, D \rangle$

- tout élément c figurant dans G vérifie $c < x$
- tout élément c figurant dans D vérifie $c > x$



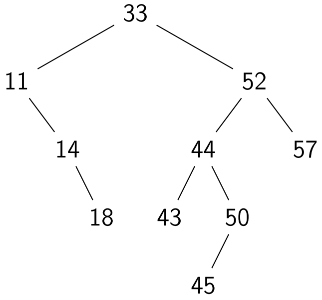
AVL-arbres (Adelson - Velskii et Landis, 1962)



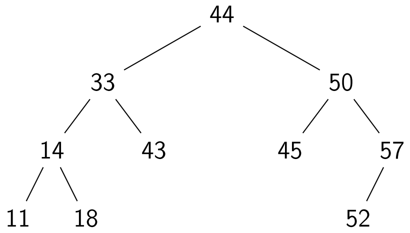
Définition. Un AVL-arbre est un arbre binaire de recherche *équilibré*, i.e. tel que pour tout noeud $p = \langle x, G, D \rangle$,

$$|hauteur(G) - hauteur(D)| \leq 1$$

AVL-arbres



ABR non équilibré



ABR respectant la propriété d'équilibre

AVL-arbres

Propriété : la hauteur h d'un AVL-arbre de taille n vérifie

$$h = \mathcal{O}(\log n)$$

AVL-arbres

Propriété : la hauteur h d'un AVL-arbre de taille n vérifie

$$h = \mathcal{O}(\log\ n)$$

Preuve. On note $s_{min}(h)$ la taille minimale d'un AVL de hauteur h.
$s_{min}(0) = 0$ et $s_{min}(1) = 1$. Si $h \geq 2$ alors

$$s_{min}(h) = 1 + s_{min}(h-1) + s_{min}(h-2)$$

AVL-arbres

Propriété : la hauteur h d'un AVL-arbre de taille n vérifie

$$h = \mathcal{O}(\log\ n)$$

Preuve. On note $s_{min}(h)$ la taille minimale d'un AVL de hauteur h.
$s_{min}(0) = 0$ et $s_{min}(1) = 1$. Si $h \geq 2$ alors

$$s_{min}(h) = 1 + s_{min}(h-1) + s_{min}(h-2)$$

$n^{\text{ième}}$ nombre de Fibonacci : $F_n = F_{n-1} + F_{n-2} \simeq \frac{\varphi^n}{\sqrt{5}}$
$F_0 = 0,\ F_1 = 1$, et $\varphi = \frac{1+\sqrt{5}}{2}$ le *nombre d'or*

AVL-arbres

Propriété : la hauteur h d'un AVL-arbre de taille n vérifie

$$h = \mathcal{O}(\log\ n)$$

Preuve. On note $s_{min}(h)$ la taille minimale d'un AVL de hauteur h. $s_{min}(0) = 0$ et $s_{min}(1) = 1$. Si $h \geq 2$ alors

$$s_{min}(h) = 1 + s_{min}(h-1) + s_{min}(h-2)$$

n ième nombre de Fibonacci : $F_n = F_{n-1} + F_{n-2} \simeq \frac{\varphi^n}{\sqrt{5}}$

$F_0 = 0,\ F_1 = 1$, et $\varphi = \frac{1+\sqrt{5}}{2}$ le *nombre d'or*

Donc $s_{min}(h) \geq \frac{\varphi^h}{\sqrt{5}}$ et pour un AVL de taille n et de hauteur h

$$h \leq \log_\varphi(\sqrt{5} \times n) = \mathcal{O}(\log\ n)$$

Rotations

Objectif : rétablir la propriété d'équilibre

- en préservant l'ordre infixe,
- pour un coût minimal (temps constant).

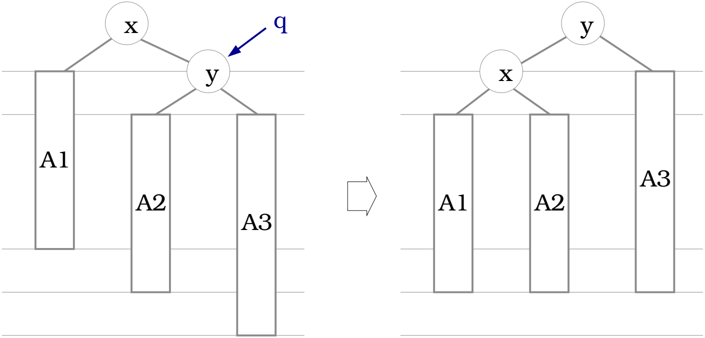
Type :

- rotation simple,
- rotation double.

Direction :

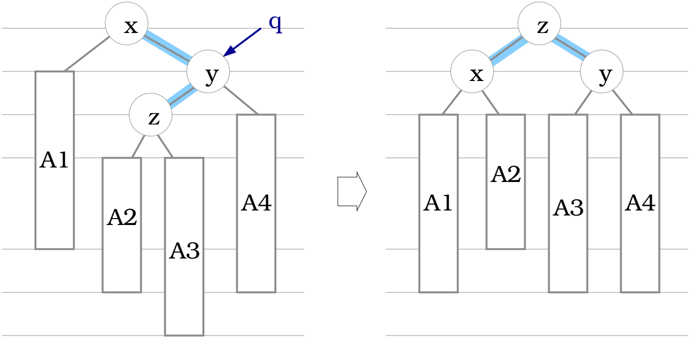
- à droite,
- à gauche.

Rotation simple (à gauche)



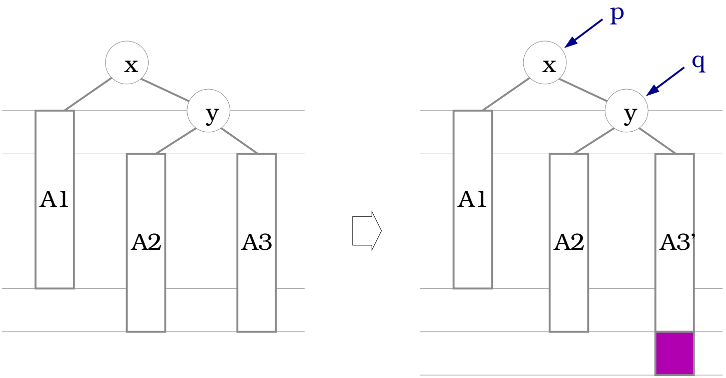
On suppose que $A1$, $A2$ et $A3$ sont équilibrés, de même que le sous-arbre enraciné en q.

Rotation double (à gauche)



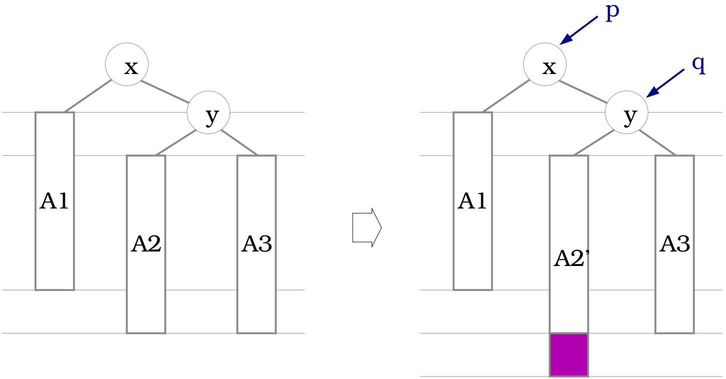
Le déséquilibre est dû au sous-arbre $A3$ (ça pourrait être $A2$),
$A1$, $A2$, $A3$ et $A4$ sont équilibrés,
ainsi que le sous-arbre enraciné en q.

Rotation simple après insertion



L'insertion d'une nouvelle valeur dans $A3$ déséquilibre l'arbre en p.
Ca penche à droite en p, ça penche à droite en q.

Rotation double après insertion



L'insertion d'une nouvelle valeur dans $A2$ déséquilibre l'arbre en p.
Ca penche à droite en p, ça penche à gauche en q.

AVL : maintien de l'équilibre

Insertion :

- insérer comme dans un ABR,
- remonter la branche en rééquilibrant éventuellement.

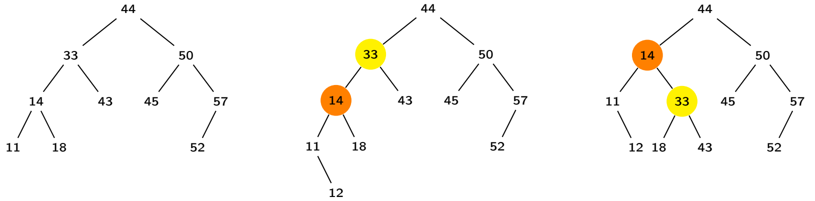
Au plus une rotation

Suppression :

- supprimer comme dans un ABR,
- remonter la branche à partir du noeud décroché, en rééquilibrant chaque fois que nécessaire.

Au plus autant de rotations que de noeuds dans la branche.

AVL : insertion exemple



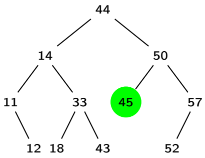
insertion de la valeur 12
déséquilibre en (33)

après rotation simple
à droite

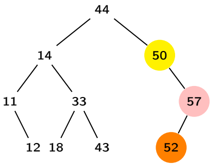
(33) est le premier noeud avec un déséquilibre rencontré si on remonte la branche à partir de (12)

Ca penche à gauche et à gauche ça penche à gauche $\Longrightarrow$ rotation simple à droite.

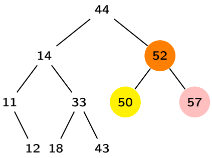
AVL : suppression exemple



suppression de 45



déséquilibre en (50)



après rotation double

(50) est le premier noeud avec un déséquilibre rencontré si on remonte la branche à partir de (50)

Ca penche à droite et à droite ça penche à gauche $\Longrightarrow$ rotation double.

C'est équilibré en (44), il n'y a pas lieu de faire d'autres rotations.