

## 9- Arbres AVL

- ▶ Arbre AVL
- ▶ Facteur d'équilibrage et rotation
- ▶ Implémentation

## Arbre AVL : Définition

### Définition :

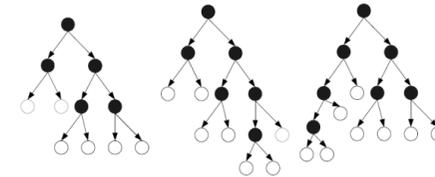
- ▶ Un arbre AVL est un arbre binaire éventuellement vide tel que la différence de hauteur entre le sous arbre gauche et le sous arbre droit d'un nœud diffère d'au plus 1.

- ▶ les arbres gauches et droits d'un sommet sont des arbres AVL

*Le nom AVL vient du nom des auteurs G.M. Adelson-Velsky et E.M.Landis (1982).*

### Propriété :

Si  $n$  est le nombre de nœuds d'un arbre AVL alors sa hauteur est  $\log_2(n)$



## Facteur d'équilibrage et rotation

Chaque nœud contient un entier appelé *facteur d'équilibrage* qui permet de déterminer si il est nécessaire de rééquilibrer l'arbre.

### Définition :

Soit  $s$  un sommet ayant pour *sous arbre gauche* (resp. *droit*)  $G_s$  (resp.  $D_s$ ). Le facteur d'équilibrage  $eq(s)$  du sommet  $s$  est défini par :

$$eq(s) = h(D_s) - h(G_s)$$

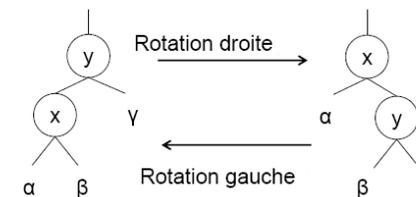
avec  $h(NIL) = 0$  et si  $A$  est un arbre  $h(A) = \text{hauteur}(A) + 1$ .

Le facteur d'équilibrage d'un nœud d'un arbre AVL vaut 0, 1 ou -1. Lors d'insertion ou suppression, l'arbre peut se *déséquilibrer* (valeur 2 ou -2), on utilise alors des rotations pour rééquilibrer l'arbre.

## Facteur d'équilibrage et rotation

### Définition :

Une rotation droite autour du sommet  $y$  d'un arbre binaire de recherche consiste à faire descendre le sommet  $y$  et à faire remonter son fils gauche  $x$  sans invalider l'ordre des éléments. L'opération inverse s'appelle rotation gauche autour du sommet  $x$ .



## Facteur d'équilibrage et rotation

Dans la rotation seuls les facteur d'équilibrage de X et Y sont modifiés.  
Notons  $eq'(X)$  et  $eq'(Y)$  les facteurs d'équilibrage après rotation.

**Propriété :** Après une rotation droite autour du sommet Y, on a :  
 $eq'(X) = eq(X) + 1 + \max(eq'(Y), 0)$   
 $eq'(Y) = eq(Y) + 1 - \min(eq(X), 0)$

**Propriété :** Après une rotation gauche autour du sommet X, on a :  
 $eq'(X) = eq(X) - 1 - \max(eq(Y), 0)$   
 $eq'(Y) = eq(Y) - 1 + \min(eq(X), 0)$

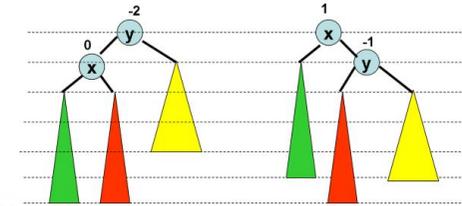
## Opérations d'insertion

Les opérations d'insertion et suppression sont celles d'un arbre binaire de recherche dans lesquelles on ajoute la gestion du facteur d'équilibrage.

Insertion : L'insertion aboutit à la création d'une feuille f.  
Considérons le premier ancêtre y de cette feuille qui viole la condition AVL, on a  $eq(y) = -2$  ou  $eq(y) = 2$  et ces deux cas sont symétriques.

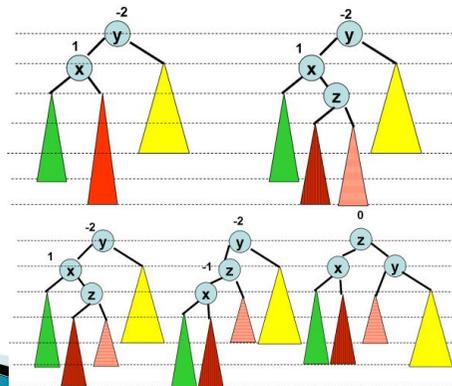
Supposons que  $eq(y) = -2$  et soit x le fils gauche de y. On a deux cas.

- soit  $eq(x) = -1$  ou 0, on effectue une rotation droite,



## Opérations d'insertion

- soit  $eq(x) = 1$ , alors x a un sous arbre droit de racine z, qui a deux sous arbres. On effectue une rotation gauche autour de x puis une rotation droite autour de y.



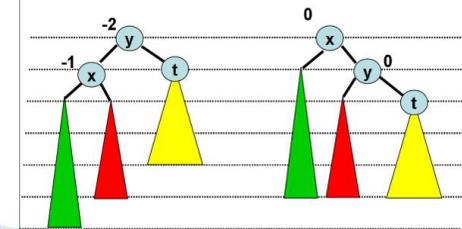
## Opérations de suppression

- La suppression aboutit à la suppression d'une feuille .
- Considérons le premier ancêtre y de cette feuille qui viole la condition AVL, on a :  
 $eq(y) = -2$  ou  $eq(y) = 2$

et ces deux cas sont symétriques.

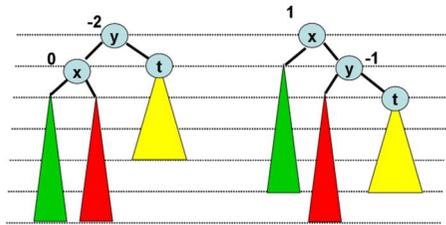
Supposons que  $eq(y) = -2$  Soit x le fils gauche de y et t le fils droit de y. La feuille a été supprimée dans le sous arbre de racine t. On a trois cas :

- soit  $eq(x) = -1$ , on effectue une rotation droite,



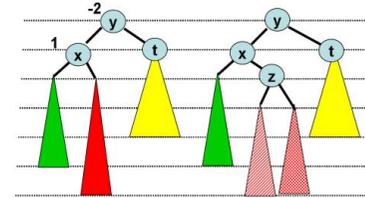
## Opérations de suppression

- soit  $eq(x)=0$ , on effectue aussi une rotation droite,



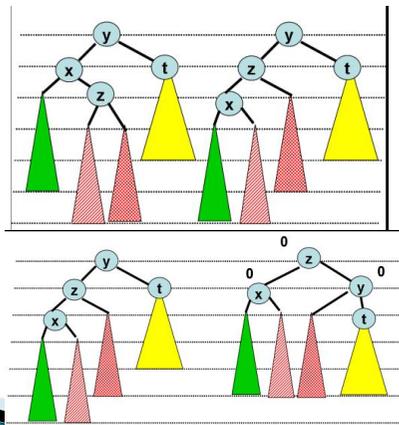
## Opérations de suppression

- soit  $eq(x)=1$ , alors x a un sous arbre droit de racine z, qui a deux sous arbres.



## Opérations de suppression

- On effectue une rotation gauche autour de x puis une rotation droite autour de y.



## Opérations de suppression

- La hauteur du nouveau sous arbre a même hauteur que le sous arbre de départ si le facteur d'équilibrage de la racine est 0.
- Dans ce cas, il suffit donc d'une opération d'équilibrage pour que l'arbre soit AVL.
- Dans les autres cas il faut recommencer la même opération en remontant dans l'arbre.
- Le cas  $eq(y)=2$  est obtenue par symétrie.
- Figures complètes [ici](#)

## Implémentation

L'intérêt du facteur d'équilibrage est que cette information nécessite 2 bits. Néanmoins les algorithmes permettant la gestion de eq sont délicats. Dans la suite, une cellule pour un arbre AVL doit contenir un champ supplémentaire qui est la hauteur.

```
type celluleAVL=structure
  info:objet;
  hauteur:entier;
  gauche:sommet;
  droit:sommet;
  père:sommet;
finstructure
sommetAVL=^celluleAVL;
```

On accède à ce nouveau champ par les deux primitives suivantes :

```
fonction getHauteur(ref S:sommet):entier;
/* 1 pour une feuille; 0 si NIL/*
fonction setHauteur(ref S:sommet; val h:entier):entier;
/* 1 pour une feuille; 0 si NIL/*
```

## Implémentation

Les fonctions implémentant les rotations sont immédiates.

```
fonction rotationDroite(ref y:sommet):vide;
var x,p:sommet;
debut
  x=filGauche(y);
  p=pere(y);
  si p!=NIL alors
    si y=filGauche(p) alors
      p^.gauche=x;
    sinon
      p^.droit=x;
    fin
  fin
  x^.pere=p;
  y^.pere=x;
  y^.gauche=x^.droit;
  x^.droit=y;

setHauteur(y,max(getHauteur(filGauche(y)),getHauteur(filDroit(y))+1);
setHauteur(x,max(getHauteur(filGauche(x)),getHauteur(filDroit(x))+1);
fin
```

## Implémentation

Les fonctions implémentant les rotations sont immédiates.

```
fonction rotationGauche(ref x:sommet):vide;
var y,p:sommet;
debut
  y=filDroit(x);
  p=pere(x);
  si p!=NIL alors
    si x=filGauche(p) alors
      p^.gauche=y;
    sinon
      p^.droit=y;
    fin
  fin
  y^.pere=p;
  x^.pere=y;
  x^.droit=y^.gauche;
  y^.gauche=x;

setHauteur(x,max(getHauteur(filGauche(x)),getHauteur(filDroit(x))+1);
setHauteur(y,max(getHauteur(filGauche(y)),getHauteur(filDroit(y))+1);
fin
```

## Implémentation

Les fonctions d'insertion et suppression doivent prendre en compte le calcul du facteur d'équilibrage ainsi que le maintien de cet équilibre.

```
fonction ajouter(ref x:sommet, val e:objet):vide;
var s:sommet;
debut
  si e ≤ getValeur(x) alors
    s=filGauche(x);
    si s==NIL alors
      ajouterFilsGauche(x,e);
      setHauteur(filGauche(x),1);
      equilibreAprèsInsertion(filGauche(x));
    sinon
      ajouter(s,e);
    fin
  sinon
    s=filDroit(x);
    si s==NIL alors
      ajouterFilsDroit(x,e);
      setHauteur(filDroit(x),1);
      equilibreAprèsInsertion(filDroit(x));
    sinon
      ajouter(s,e);
    fin
  fin
fin
```

## Implémentation

```

fonction supprimer(ref x:sommet):booléen;
var p,f,y:sommet;
début
  si estFeuille(x) alors
    p=pere(x);
    si filsGauche(p)==x alors
      supprimerFilsGauche(p)
      setHauteur(p,getHauteur(filsDroit(p)+1))
    sinon
      supprimerFilsDroit(p)
      setHauteur(p,getHauteur(filsGauche(p)+1))
    finsi
  fin
  si equilibreAprèsSuppression(p)
  sinon
    f=filsDroit(x);
    si f!=NIL
      y=cherchePlusPetit(f);
    sinon
      f=filsGauche(x);
      y=cherchePlusGrand(f);
    finsi
    setValeur(x,getValeur(y));
    supprimer(y);
  fin
fin

```

## Implémentation

```

fonction equilibreAprèsInsertion(ref x:sommet):vide;
var eq:entier;
var s,p:sommet;
debut
  eq=0;
  p=x;
  tantque pere(p)!=NIL et eq!=2 et eq!=2 faire
    s=p;
    p=pere(s);
    setHauteur(p,max(getHauteur(filGauche(p)),getHauteur(filsDroit(p))+1));
    eq=getHauteur(filsDroit(p))-getHauteur(filsGauche(p));
  fintantque
  si eq==2 ou eq==-2 alors
    equilibreUnSommet(p,s);
  fin
fin

```

## Implémentation

```

fonction equilibreUnSommet(ref p,s:sommet):vide;
début
  si s==filsGauche(p) alors
    si getEquilibre(p)==-2 alors
      si getEquilibre(s)<1 alors
        rotationDroite(p);
      sinon
        rotationGauche(s);
        rotationDroite(p);
      finsi
    fin
  sinon
    si getEquilibre(p)==2 alors
      si getEquilibre(s)>-1 alors
        rotationGauche(p)
      sinon
        rotationDroite(s);
        rotationGauche(p);
      finsi
    fin
  fin
fin

```

## Implémentation

Dans le cas de suppression, on remonte à partir de la feuille supprimée dans l'arbre et de la même manière on détecte les nœuds déséquilibrés. Il est possible que plusieurs rotations soient nécessaires. En effet un déséquilibre peut en entrainer un autre.

```

fonction equilibreAprèsSuppression(ref x:sommet):vide;
var eq:entier;
var s,p:sommet;
début
  eq=1;
  p=x;
  p=pere(s);
  tantque p!=NIL et eq!=0 faire
    s=p;
    p=pere(p);
    setHauteur(p,max(getHauteur(filGauche(p)),getHauteur(filsDroit(p))+1));
    eq=getHauteur(filsDroit(p))-getHauteur(filsGauche(p));
    equilibreUnSommet(p,s);
  fintantque
  fin

```