Structures de données avancées :

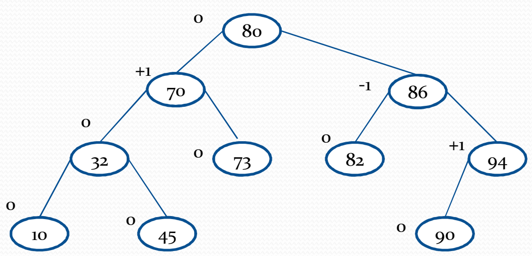
Arbres AVL

Pr ZEGOUR DJAMEL EDDINE
Ecole Supérieure d'Informatique (ESI)
http://zegour.esi.dz
email: d_zegour@esi.dz

Les arbres AVL

Arbres AVL

Un *arbre AVL* **est un** *arbre de recherche binaire équilibré*

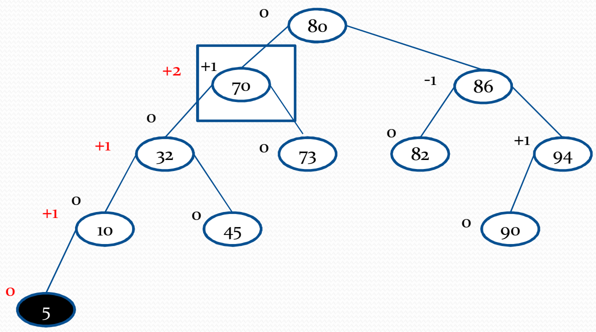


| Profondeur(fg(n))– Profondeur(fd(n)) | <= 1

Ajouter un champ balance (facteur d'équilibrage) au niveau de chaque noeud

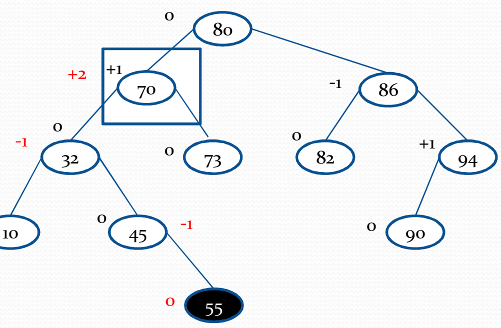
Les arbres AVL

Arbres AVL (*Cas de déséquilibre*)



Les arbres AVL

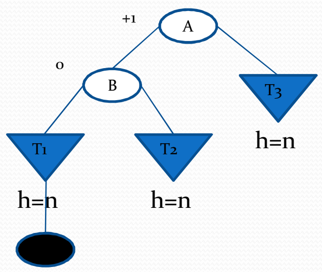
Arbres AVL (*Cas de déséquilibre*)



Les arbres AVL

Arbres AVL (*Techniques d'équilibrage*)

Examinons un sous arbre de racine le plus jeune antécédent qui devient non équilibré suite à une insertion



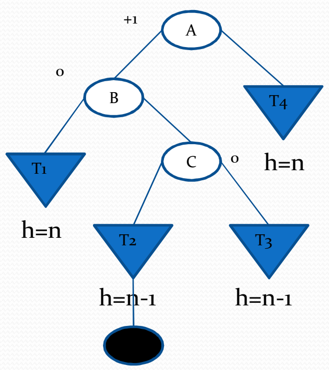
Cas où le facteur d'équilibrage est +1

Le nouveau nœud est inséré dans le sous arbre gauche de B. Donc f(B) devient 1 et f(A) devient 2

Les arbres AVL

Arbres AVL (*Techniques d'équilibrage*)

Examinons un sous arbre de racine le plus jeune antécédent qui devient non équilibré suite à une insertion



Cas où le facteur d'équilibrage est +1

Le nouveau nœud est inséré dans le sous arbre droit de B. f(B) devient -1 et f(A) devient 2.

Les arbres AVL

Arbres AVL (*Techniques d'équilibrage)*

Transformer l'arbre de telle sorte que

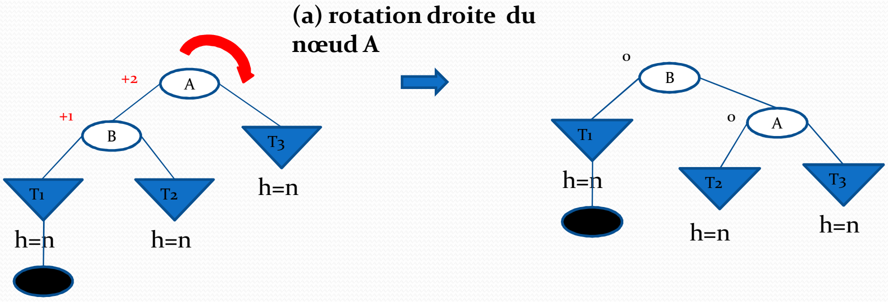
l'inordre soit préservé

l'arbre transformé soit équilibré

Les arbres AVL

Arbres AVL (*Techniques d'équilibrage)*

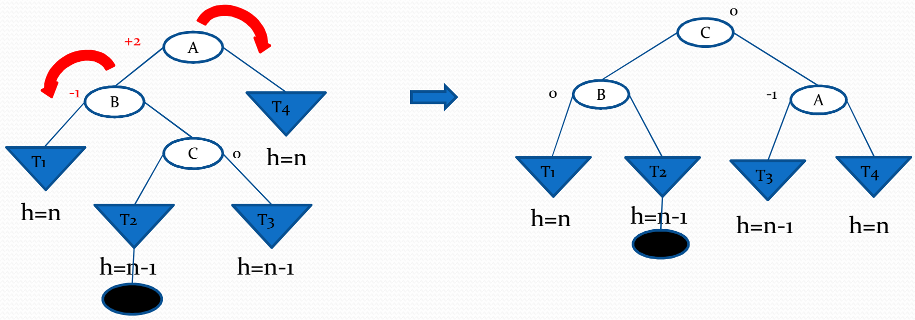
(a) rotation droite du nœud A



Les arbres AVL

Arbres AVL (*Techniques d'équilibrage*)

(b) rotation gauche du nœud B suivie par une rotation droite du nœud A



Les arbres AVL

Arbres AVL (*Algorithme d'insertion*)

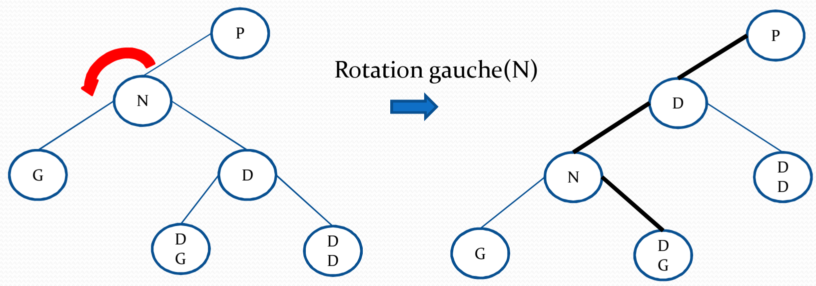
La première partie de l'algorithme consiste à insérer la clé dans l'arbre sans tenir compte du facteur d'équilibrage

Elle garde aussi la trace du plus jeune antécédent, soit Y qui devient non équilibré

La deuxième partie fait la transformation à partir de Y

Les arbres AVL

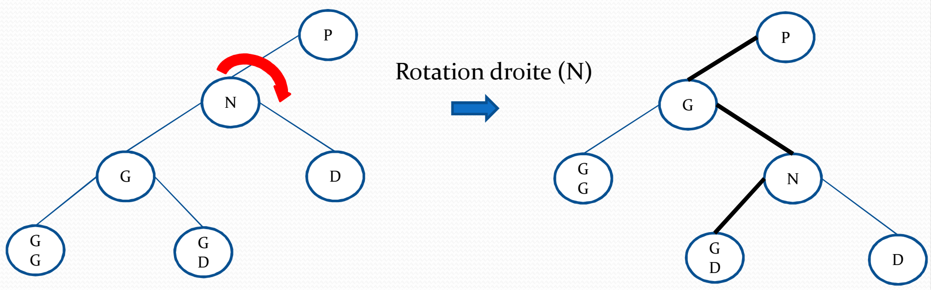
Arbres AVL (*Rotation gauche*)



AFF_FD(N, DG) AFF_FG(D, N)) AFF_FG(Parent, D)

Les arbres AVL

Arbres AVL (*Rotation droite*)



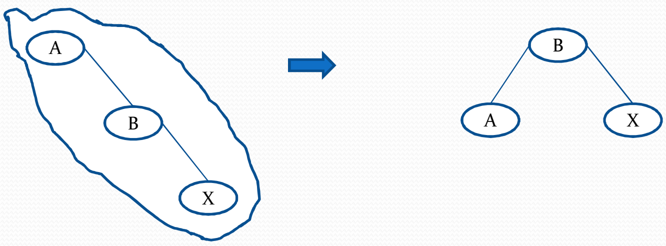
AFF_FG(N, GD) AFF_FD(G, N)) AFF_FD(Parent, G)

Les arbres AVL

<u>Arbres AVL (*Exemples*)</u>

Insérons la séquence : A, B, X, L, M, C, D, E, H, R, F **dans un arbre AVL.**

Insertion A, B, X

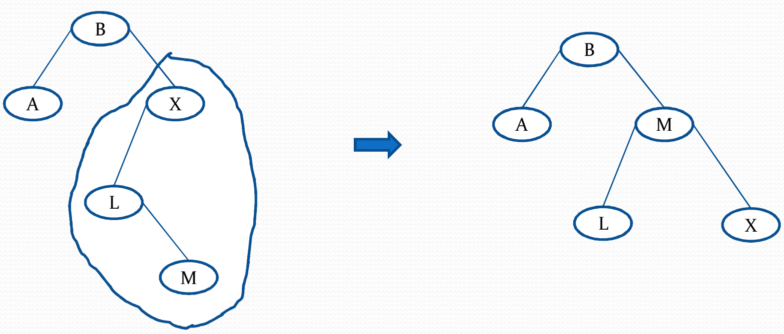


Les arbres AVL

<u>Arbres AVL (*Exemples*)</u>

Insérons la séquence : A, B, X, L, M, C, D, E, H, R, F **dans un arbre AVL.**

Insertion L, M

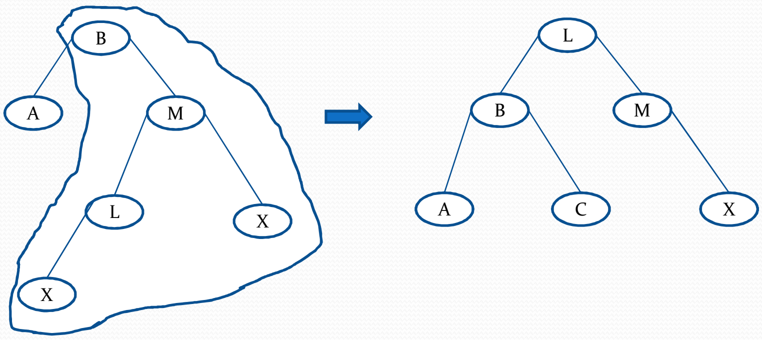


Les arbres AVL

Arbres AVL (*Exemples*)

Insérons la séquence : A, B, X, L, M, C, D, E, H, R, F **dans un arbre AVL.**

Insertion C

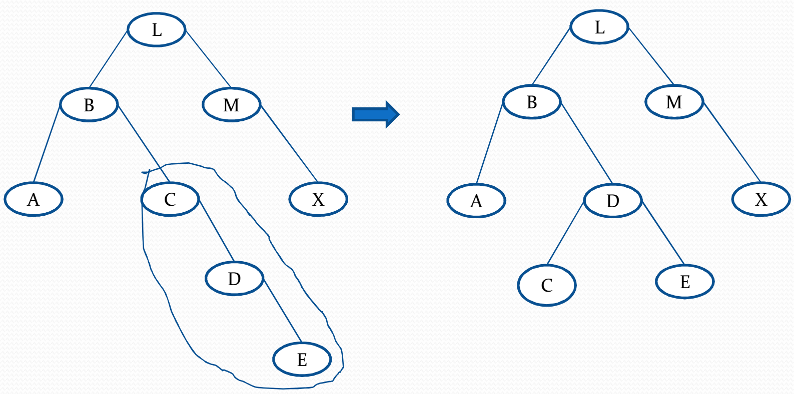


Les arbres AVL

Arbres AVL (*Exemples*)

Insérons la séquence : A, B, X, L, M, C, D, E, H, R, F **dans un arbre AVL.**

Insertion D, E

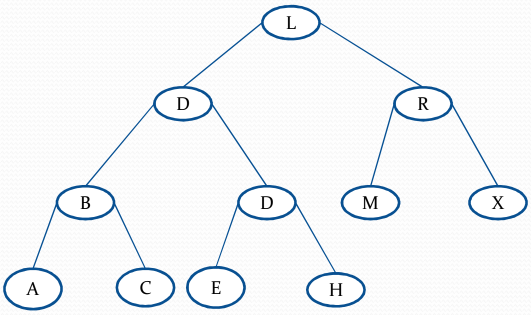


Les arbres AVL

Arbres AVL (*Exemples*)

Insérons la séquence : A, B, X, L, M, C, D, E, H, R, F **dans un arbre AVL.**

Insertion H, R, F



Les arbres AVL

Arbres AVL (*Suppression*)

Étape 1 : comme dans un arbre de recherche binaire ordinaire
Étape 2 : mettre à jour les balances

Cas où la balance d'un nœud A devient +2
→ Le fils gauche B de A doit exister

Les cas suivants peuvent se présenter

a) B a une balance égale à + 1
b) B a une balance égale à – 1
c) B a une balance égale à 0

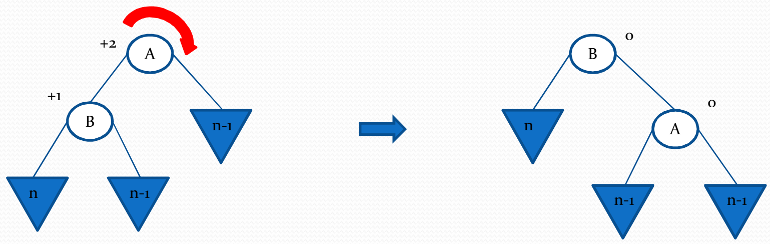
Même traitement symétrique dans le cas où la balance d'un nœud A devient -2

Traitement peut continuer en cascade

Les arbres AVL

<u>Arbres AVL (*Suppression*)</u>

B a une balance égale à + 1

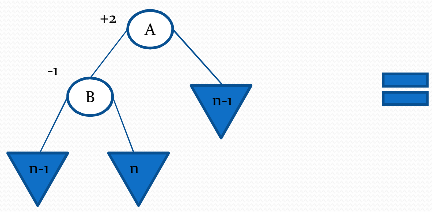


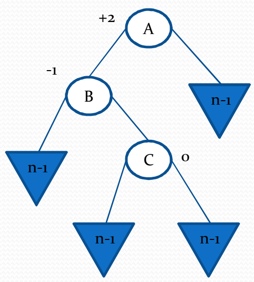
Les arbres AVL

<u>Arbres AVL (*Suppression*)</u>

B a une balance égale à -1

Cas Balance (C)= 0



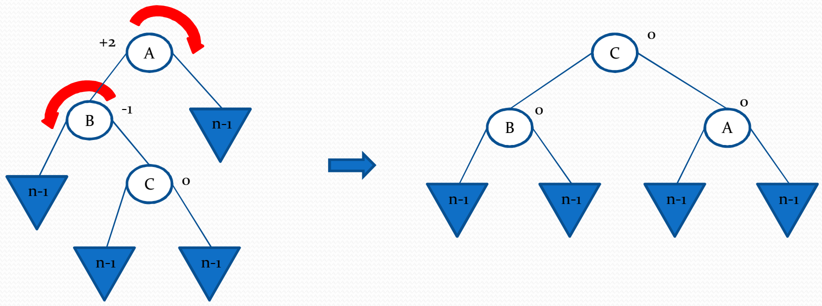


B a donc un fils à sa droite, soit C.

Les arbres AVL

Arbres AVL (*Suppression*)

B a une balance égale à -1 , C son fils droit avec Balance(C)=0

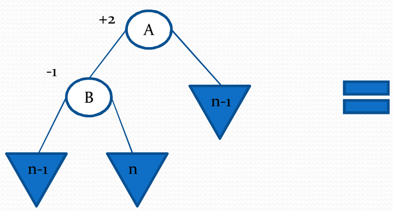


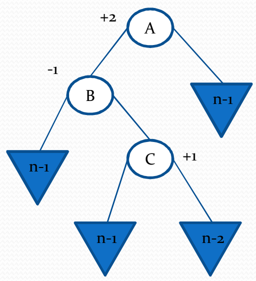
Les arbres AVL

Arbres AVL (*Suppression*)

B a une balance égale à -1

Balance (C)= +1



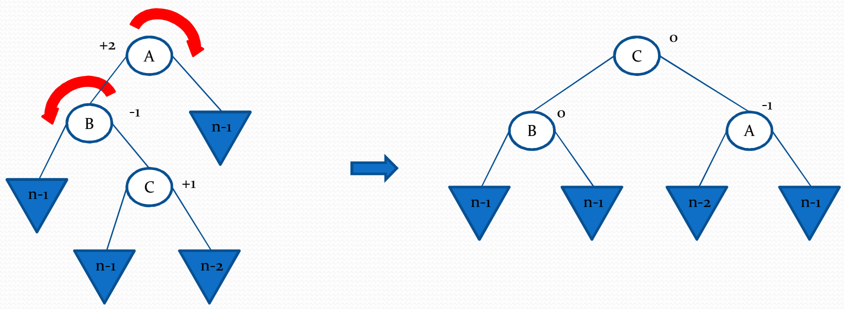


B a donc un fils à sa droite, soit C.

Les arbres AVL

Arbres AVL (*Suppression*)

B a une balance égale à -1 , C son fils droit avec Balance(C)=+1

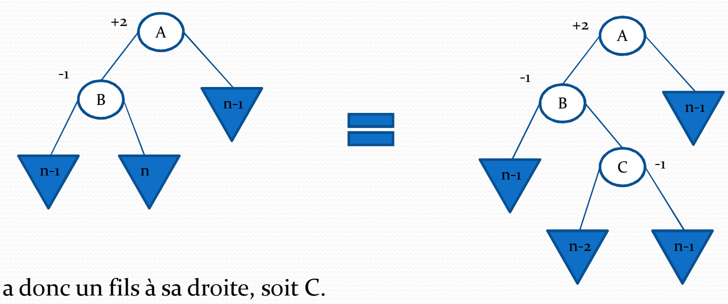


Les arbres AVL

Arbres AVL (*Suppression*)

B a une balance égale à -1

Balance (C)= -1

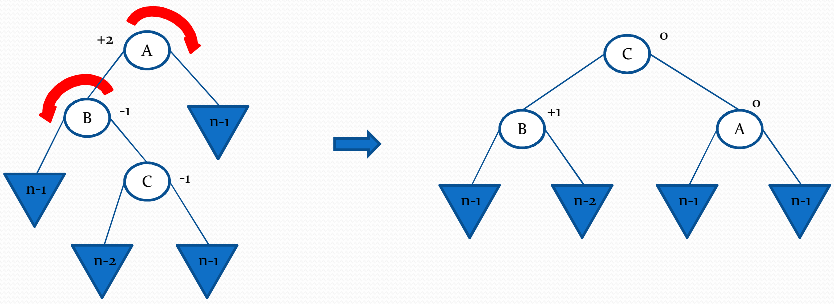


B a donc un fils à sa droite, soit C.

Les arbres AVL

Arbres AVL (*Suppression*)

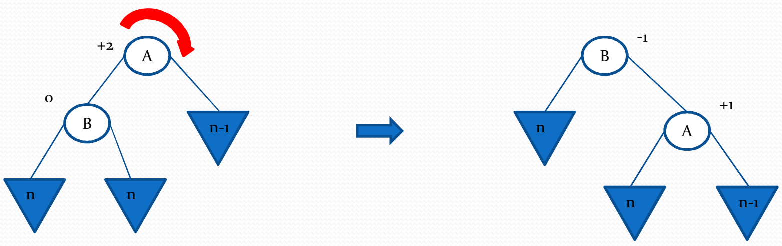
B a une balance égale à -1 , C son fils droit avec Balance(C)=-1



Les arbres AVL

Arbres AVL (*Suppression*)

B a une balance égale à 0



Les arbres AVL

Arbres AVL (*Analyse théorique*)

*la profondeur maximale d'un arbre binaire équilibré est 1.44*Log$_2$n*

La recherche dans un tel arbre *n'exige jamais plus de 44% de plus de comparaisons que pour un arbre binaire complet*

Operations de maintenance :

- **Restructuration** = *1 rotation ou double rotation*
- **Insertion :** *au plus 1 restructuration*
- **suppression :** *au plus Log2 (N) restructurations*