

# Structures de données avancées : *B arbres*

Pr ZEGOUR DJAMEL EDDINE  
Ecole Supérieure d'Informatique (ESI)  
[www.zegour.univ.dz](http://www.zegour.univ.dz)  
email: [d\\_zegour@esi.dz](mailto:d_zegour@esi.dz)

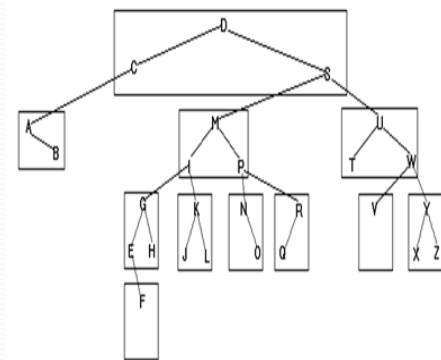
## B-arbres

### *Introduction aux B-arbres*

- Problème avec les ARM: comment garder l'arbre paginé équilibré?
- Si l'ensemble des clés est connu à l'avance c'est simple.
- Si les clés arrivent de façon aléatoire, impossible de garder l'ARM équilibré

## B-arbres

### Introduction aux B-arbres



Exemple : insertion dans un ARM d'ordre 4;

Séquence à insérer :

C S D T A M P I B W N G U R  
K E H O L J Y Q Z F X V

Revient à insérer un ensemble de clés dans un arbre AVL paginé :

## B-arbres

### Définition

- Technique plus complexe mais garantit l'équilibrage de l'arbre (donc recherche plus rapide)
- En plus, chaque nœud de l'arbre (sauf la racine) est au moins rempli à moitié (donc peu d'espace perdu).
- ==> reste encore très utilisée dans les systèmes de fichier actuels.

## B-arbres

### Définition

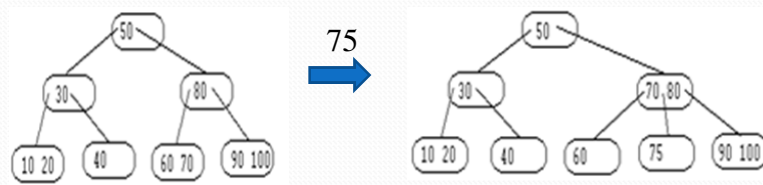
Un arbre B d'ordre  $n$  est tel que :

- ❑ la racine a au moins 2 fils
- ❑ chaque nœud, autre que la racine, a entre  $n/2$  et  $n$  fils
- ❑ tous les nœuds feuilles sont au même niveau

Remarque : B pour Bayer / Boeing / Balanced

## B-arbres

### Illustration du mécanisme d'insertion (ordre = 3)

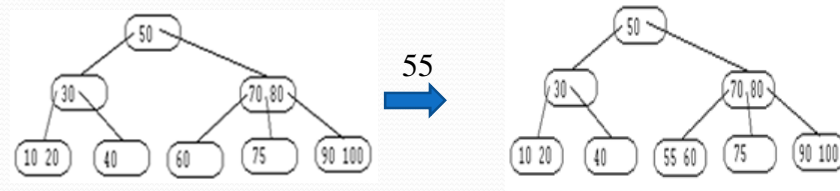


**Construction Bottum-up ( équilibrage garanti )**

Les éclatements peuvent être en cascade.

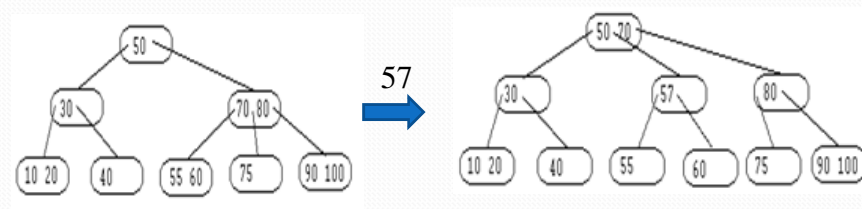
## B-arbres

*Illustration du mécanisme d'insertion (ordre = 3)*



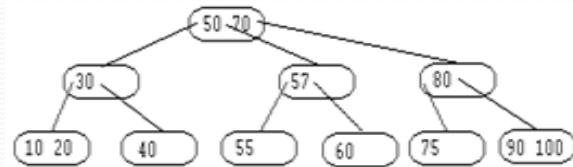
## B-arbres

*Illustration du mécanisme d'insertion (ordre = 3)*

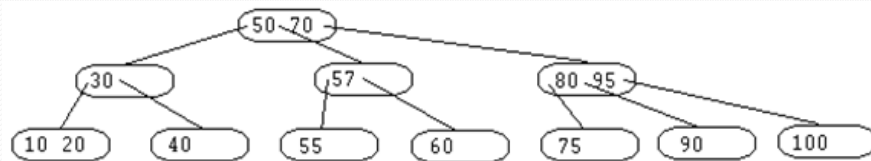


## B-arbres

*Illustration du mécanisme d'insertion (ordre = 3)*

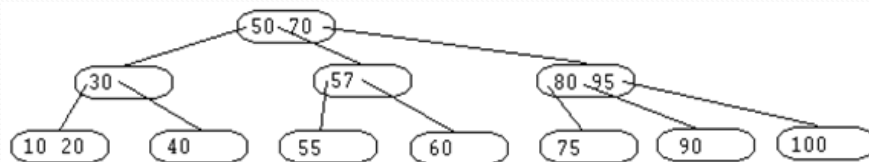


↓ 95

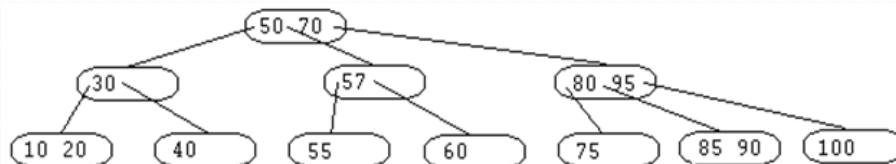


## B-arbres

*Illustration du mécanisme d'insertion (ordre = 3)*

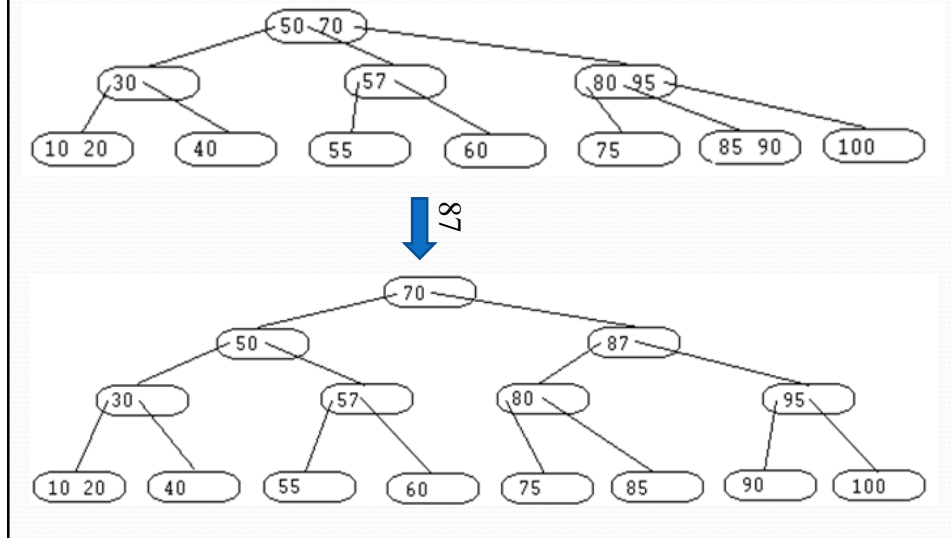


↓ 85



## B-arbres

### *Illustration du mécanisme d'insertion (ordre = 3)*



## B-arbres

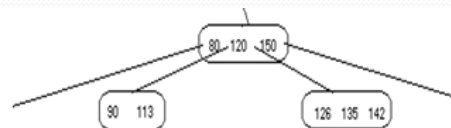
### *Suppression dans un arbre B*

- Même principe que la suppression physique dans un ARM
- En plus, si le nœud feuille qui contenait le successeur a moins de  $(n \div 2)$  clés diverses actions seront entreprises.

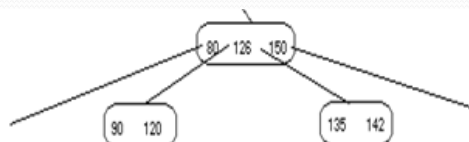
## B-arbres

### Suppression dans un arbre B

- Examiner les frères à droite et à gauche.
- Si l'un des frères (gauche ou droit) contient plus de  $n \div 2$  clés, EMPRUNT



↓ 113



Ordre=5; Max=4 données; Min = 2 données

## B-arbres

### Suppression dans un arbre B

Si les deux frères contiennent exactement  $n \div 2$  clés, FUSION



↓ 120

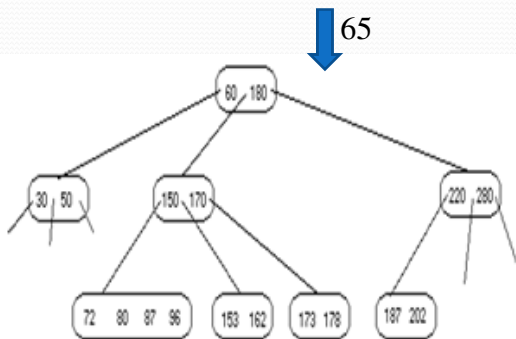


Ordre=5; Max=4 données; Min = 2 données

## B-arbres

### Suppression dans un arbre B

Cas où le père contient seulement  $n \div 2$  clés et par conséquent il n'a pas de clé à donner. → il peut emprunter de son père et frère.

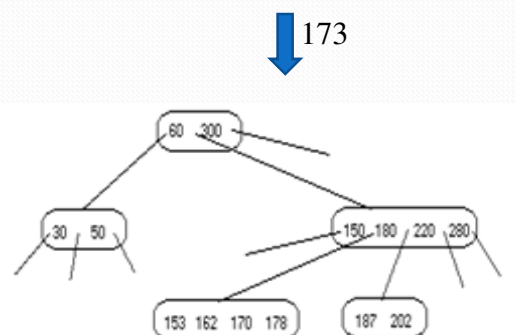


Ordre=5; Max=4 données; Min = 2 données

## B-arbres

### Suppression dans un arbre B

Les frères du père n'ont pas de clés à donner, le père et son frère peuvent aussi être concaténés et une clé est prise du grand père



Ordre=5; Max=4 données; Min = 2 données



## B-arbres

### *Suppression dans un arbre B*

- Si tous les antécédents d'un nœud et leurs frères contiennent exactement  $n \div 2$  clés, une clé sera prise de la racine (due aux concaténations en cascades).
- Si la racine avait plus d'une clé, ceci termine le processus.
- Si la racine contenait une seule clé, elle sera utilisée dans la concaténation. Le nœud racine est libéré. le niveau de l'arbre est réduit d'une unité.

## B-arbres

### *Utilisation d'un arbre B pour le stockage des données*

- Un nœud de l'arbre = un bloc sur le disque.
- A un moment donné seul un bloc est ramené en mémoire.
- Nécessite des blocs de très grande taille ( ordre de 200).
- Deux façons de ranger les articles :
  - ✓ (i) dans les nœuds avec les clés, ( Fichier = B-arbre )
  - ✓ (ii) séparément, donc pointeur additionnel vers l'information. ( B-arbre = Index )
- Si le fichier est grand on préfère de loin utiliser (ii).

## B-arbres

### *Résultats*

- Facteur de chargement de 70% pour les insertions aléatoires
- 2 à 3 accès disque pour retrouver des articles parmi des centaines de millions.
- Fichiers dynamiques ( Sans réorganisation )
- Fichiers ordonnés

## B-arbres

### *Variante : Arbres B\**

- Retarder la division quand un nœud est plein → Redistribution équitable des clés contenues dans le nœud en question et l'un de ses frères.
- Si le nœud et son frère sont tous les deux pleins, les deux nœuds sont divisés en 3. Ceci garantit un minimum de chargement à 67% et un chargement moyen de 81%.

## B-arbres

### *Variante : Arbres B+*

- Toutes les clés sont maintenues au niveau des feuilles
- Les clés sont dupliquées dans les nœuds non feuilles.
- Les articles(ou les pointeurs vers les articles) sont au niveau des feuilles.
- Les nœuds feuilles sont chaînés (ensemble des séquences)

## B-arbres

### *Variante : Arbres B+ avec expansion partielle*

- Arbre B+ avec amélioration du facteur de chargement
- Utilise le concept d'élasticité des nœuds
- Garantit un chargement moyen de 83%.