

Les arbres

Arbres m-aires - Arbres de recherche m-aire - B-arbres

D.E ZEGOUR

Ecole Supérieure d'Informatique

ESI

Les arbres m-aires

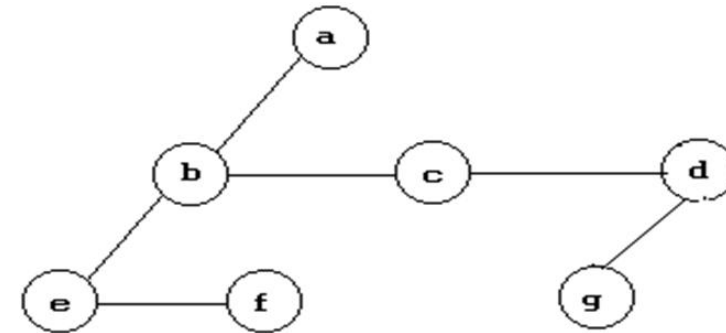
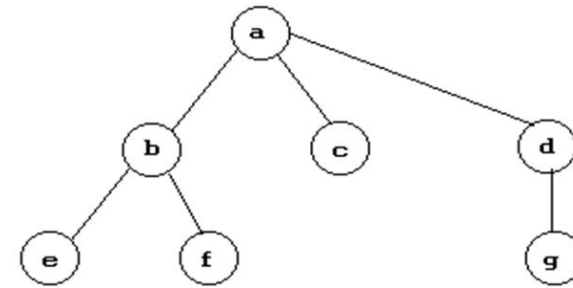
Arbre m-aire : généralisation de l'arbre binaire

Structure d'un nœud :

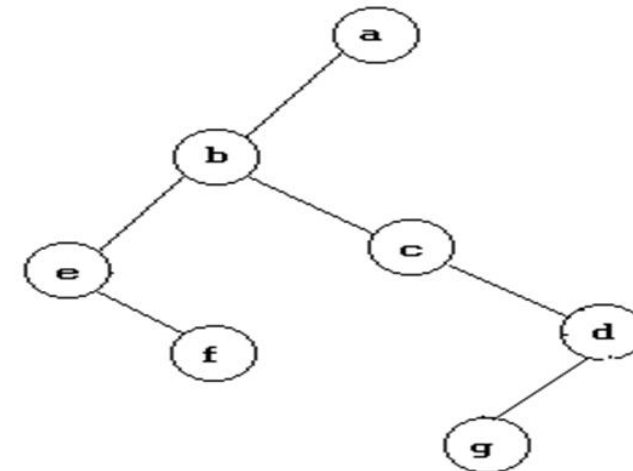
$(k_1, s_1, s_2, \dots, s_n)$

Pas de relation d'ordre

Peut se transformer en un arbre binaire



Lier les frères



Rotation 45°

Les arbres m-aires

Arbre m-aire : Machine abstraite

Creernoed(X) : Allocation d'un noeud

Liberernoed(P) : Libération d'un noeud

Fils(P, i) : i-ième fils du noeud P

Info(P) : information rattaché au noeud P

Degre(P) : Degré du noeud P

Aff_fils(P, i, Q) : Faire de Q le i-ième fils du noeud P

Aff_info(P, X) : Affecter l'information X au noeud P

Aff_degre(P, d) : Affecter le degré d au noeud P

Structure d'un nœud :

$(k_1, s_1, s_2, \dots, s_n)$

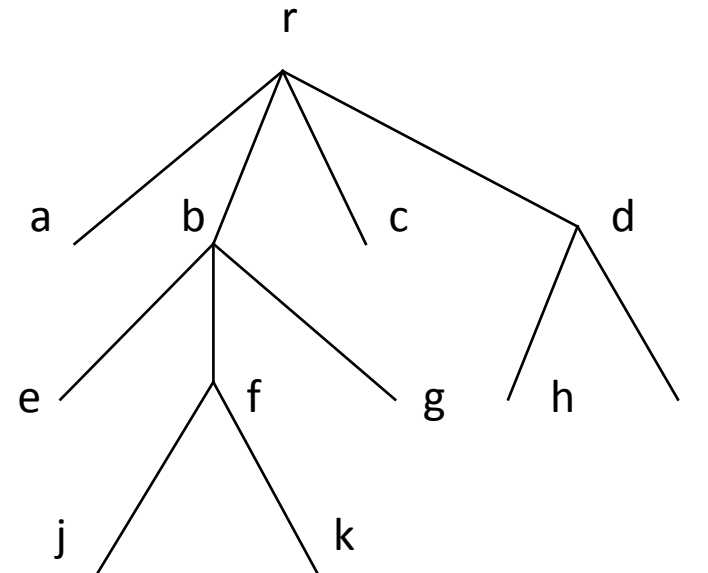
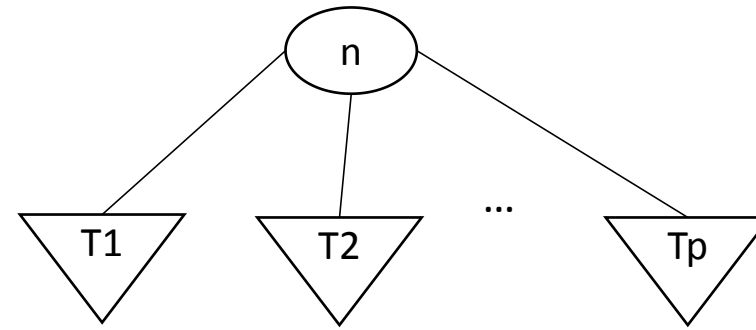
Les arbres m-aires

Arbre m-aire : Parcours

Préordre : $n \ T1 \ T2 \ \dots \ Tp$
(r a b e f j k g c d h i)

Inordre : $T1 \ n \ T2 \ \dots \ Tp$
(a r e b j f k g c h d i)

Postordre : $T1 \ T2 \ \dots \ Tp \ n$
(a e j k f g b c h i d r)



Les arbres m-aires

Arbre de recherche m-aire

Généralisation de l'arbre de recherche binaire
(Relation d'ordre)

Structure d'un nœud :

$(s_1, k_1, s_2, \dots, k_{n-1}, s_n)$

$k_1 < k_2 < \dots < k_{n-1}$

(Éléments dans s_1) $< k_1$

(Éléments dans s_j) $> k_{j-1}$ et $< k_j$

($j=2, 3, \dots, n-1$)

(Éléments dans s_n) $> k_{n-1}$

Arbre de recherche m-aire
TOP-DOWN : Tout nœud non rempli doit être une feuille.

B-arbre : Arbre de recherche m-aire équilibré

Les arbres m-aires

Arbre de recherche m-aire : Machine abstraite

Creernoead(X) : Allocation d'un noeud

Liberernoead(P) : Libération d'un noeud

Fils(P, i) : i-ième fils du noeud P

Info(P, i) : i-ième information rattaché au noeud P

Degre(P) : Degré du noeud P

Aff_fils(P, i, Q) : Faire de Q le i-ième fils du noeud P

Aff_info(P, i, X) : Affecter X comme i-ième information du noeud P

Aff_degre(P, d) : Affecter le degré d au noeud P

Structure d'un nœud :

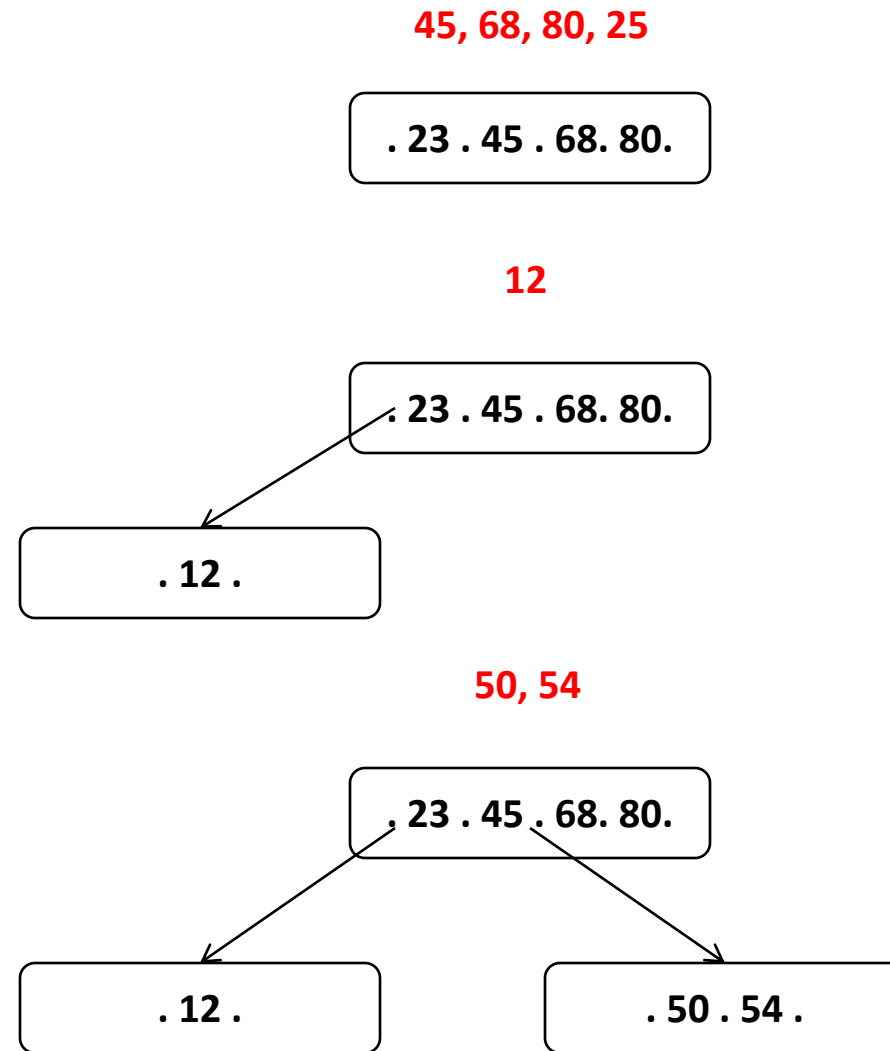
(s₁, k₁, s₂,k_{n-1},s_n)

Les arbres m-aires

Arbre de recherche TOP-DOWN

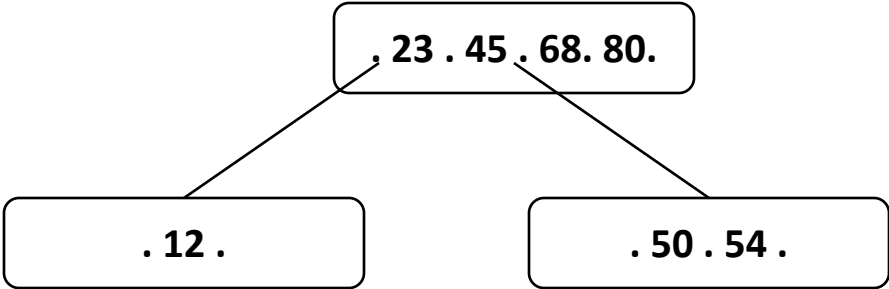
Ordre=5

Mécanisme d'insertion

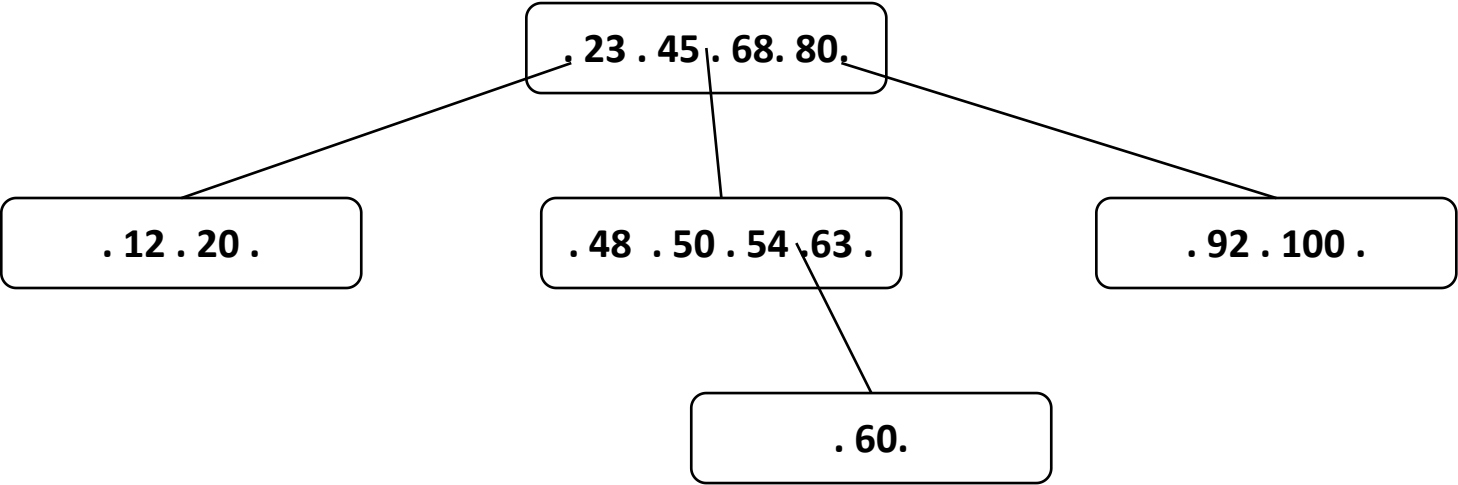


Les arbres m-aires

Arbre de recherche TOP-DOWN
Ordre=5
Mécanisme d'insertion



20, 92, 48, 63, 60, 100



Les arbres m-aires

Arbre de recherche TOP-DOWN

Mécanisme de suppression

Rechercher la donnée d à supprimer

1. Si donnée d existe sur une feuille N :

- la supprimer
- si d était seule, libérer le noeud N
- Stop

2 Si donnée d existe sur un noeud interne N:

2.a) si d a un sous arbre gauche (ou droit)non vide

- Remplacer d par son successeur (ou prédécesseur) p du noeud N'
- Poser $d:=p$ et $N:= N'$, Allera 1

2.b) si d n'a pas un sous arbre gauche (ou droit)non vide

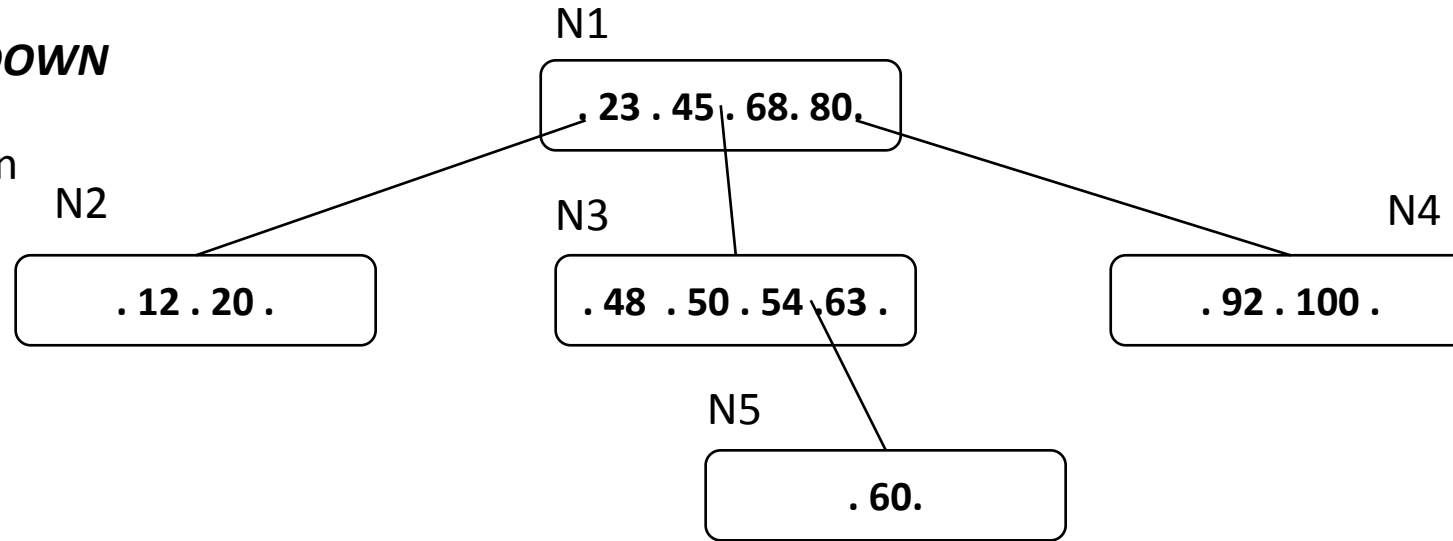
- Prendre une donnée d' du noeud N (à droite ou à gauche) qui a un sous arbre gauche ou droit (elle doit exister)
- Chercher le successeur (prédécesseur) p' de d' du noeud N'
- Supprimer d du noeud N et rajouter p' au noeud N en procédant par décalage
- $d:=p'$ et $N:= N'$, Allera 1

Les arbres m-aires

Arbre de recherche TOP-DOWN

Ordre=5

Mécanisme de suppression



Suppression 92 :

→ N4 contient 100

Suppression 80 :

→ Remplacer 80 par 92 dans N1, N4 contient 100

Supprimer 60 :

→ Libérer N5

Supprimer 45 :

→ Remplacer 45 par 48 dans N1, déplacer 60 dans N3, Libérer N5

Les arbres m-aires

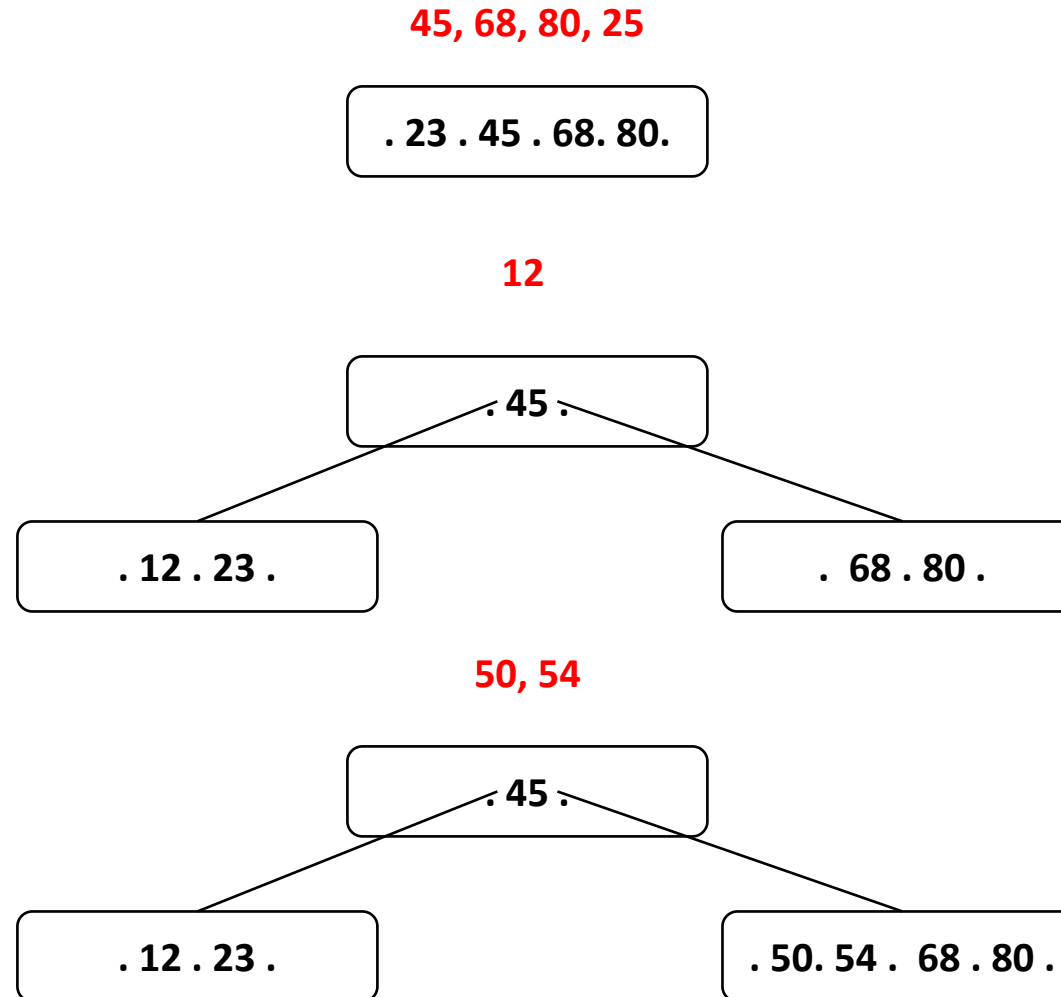
B-arbre (ordre=5)

Min=2 données; Max=4 données

Mécanisme d'insertion

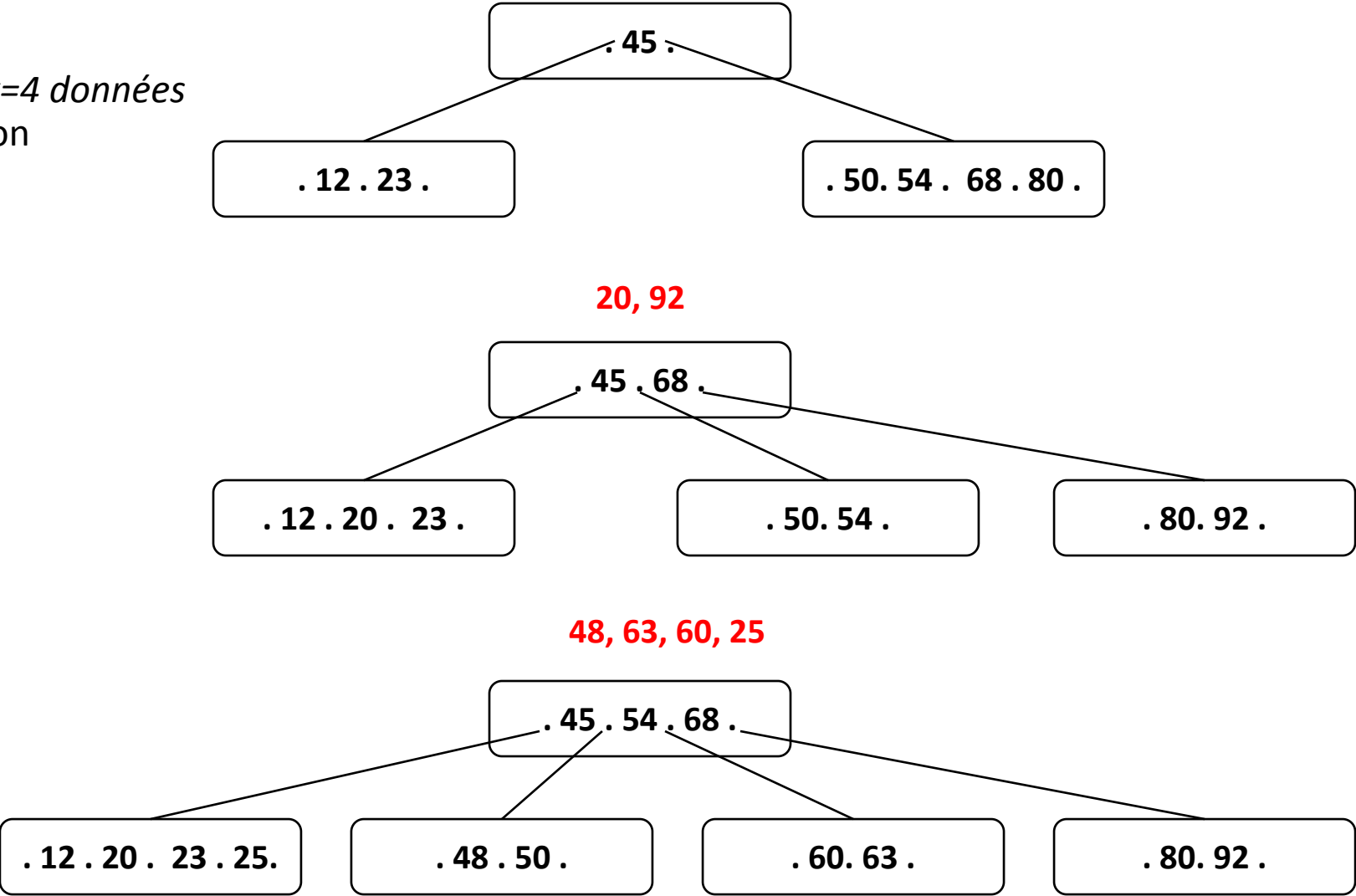
Définition (B-arbre d'ordre n)

- La racine a au moins deux fils -
- Chaque noeud (non racine) a au moins $n/2$ fils
- Toutes les feuilles sont au même niveau



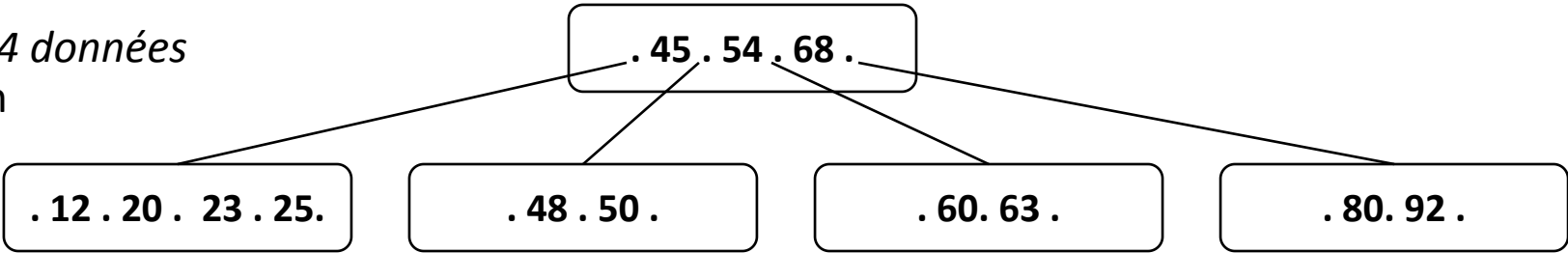
Les arbres m-aires

B-arbre (ordre=5)
Min=2 données; Max=4 données
Mécanisme d'insertion

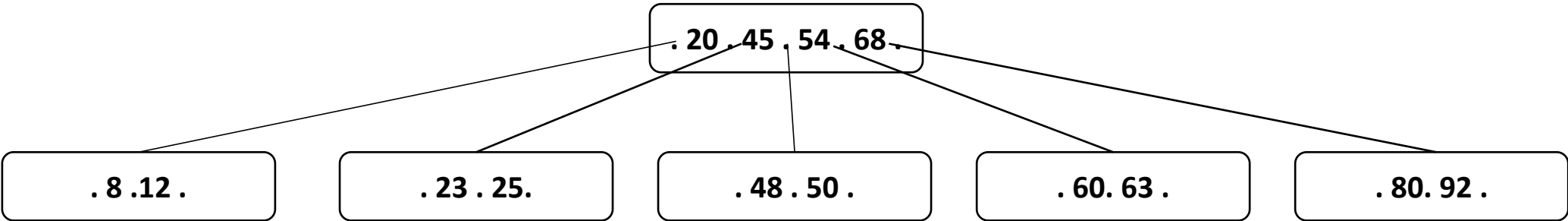


Les arbres m-aires

B-arbre (ordre=5)
Min=2 données; Max=4 données
Mécanisme d'insertion

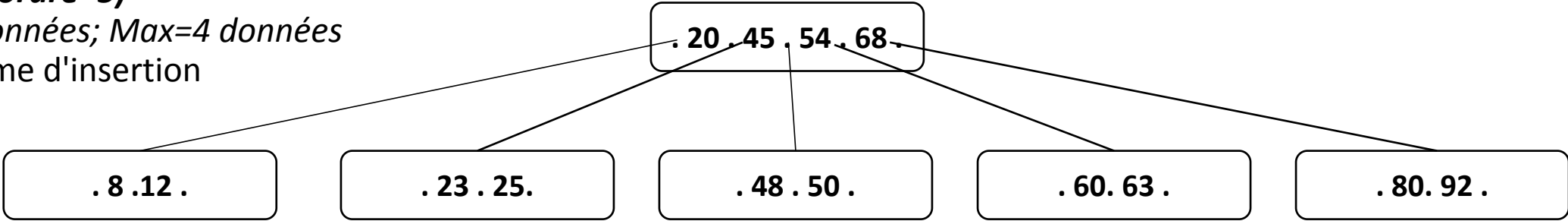


8

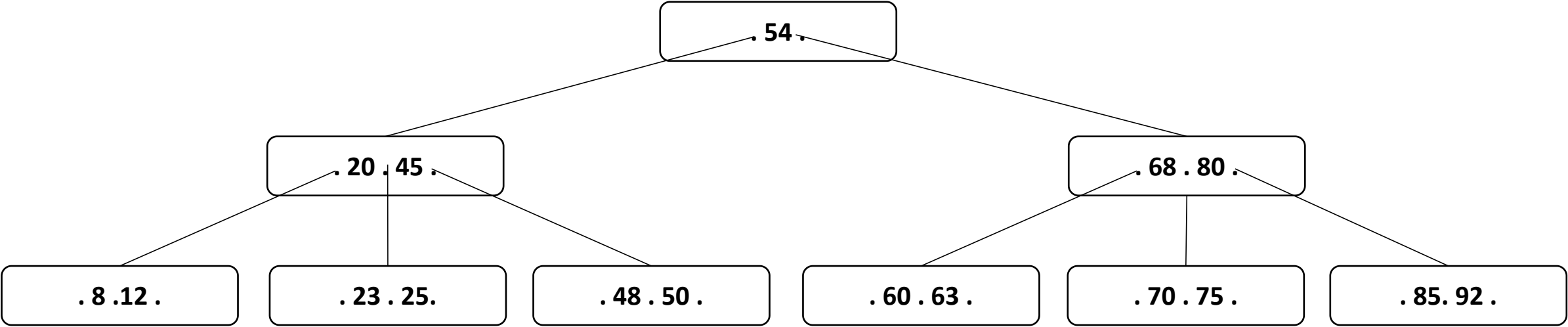


Les arbres m-aires

B-arbre (ordre=5)
Min=2 données; Max=4 données
Mécanisme d'insertion



70, 75, 85



Les arbres m-aires

Arbre de recherche m-aire : Suppression

Même principe que la suppression physique dans un ARM

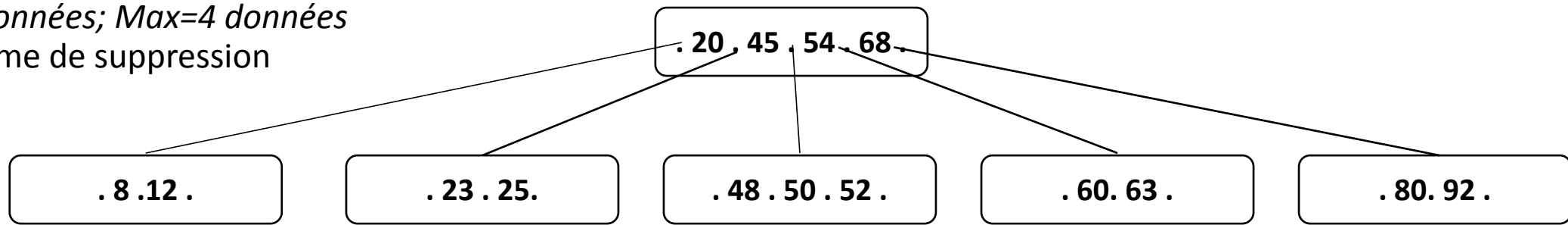
De plus, si le nœud feuille qui contenait le successeur a moins de $(n \div 2)$ données alors diverses actions seront entreprises.

Les arbres m-aires

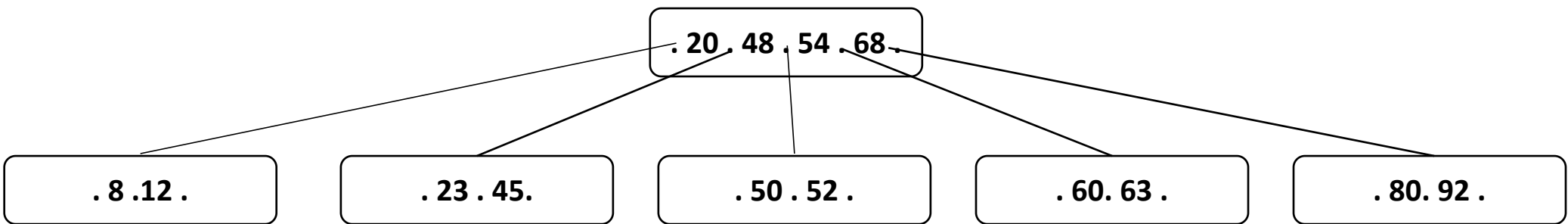
Examiner les frères à droite et à gauche.

Si l'un des frères(gauche ou droit) contient plus de $n \div 2$ clés, EMPRUNT

B-arbre (ordre=5)
Min=2 données; Max=4 données
Mécanisme de suppression



25

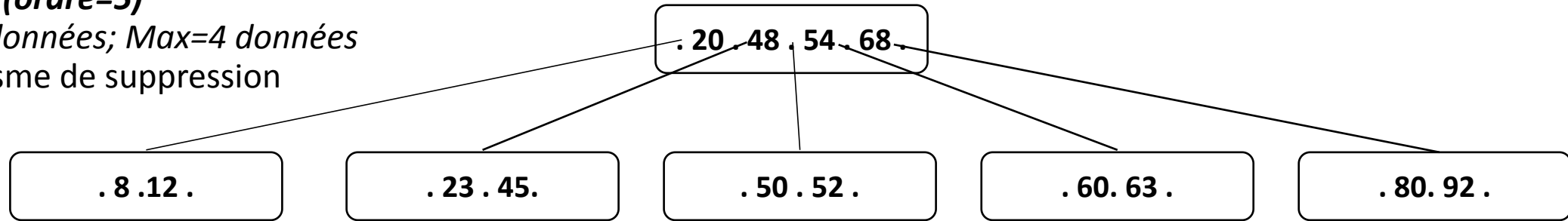


Les arbres m-aires

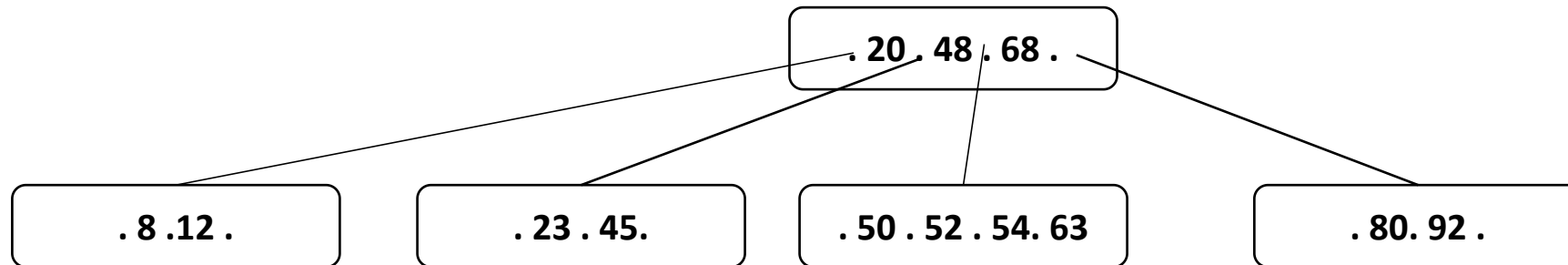
B-arbre (ordre=5)

Min=2 données; Max=4 données

Mécanisme de suppression



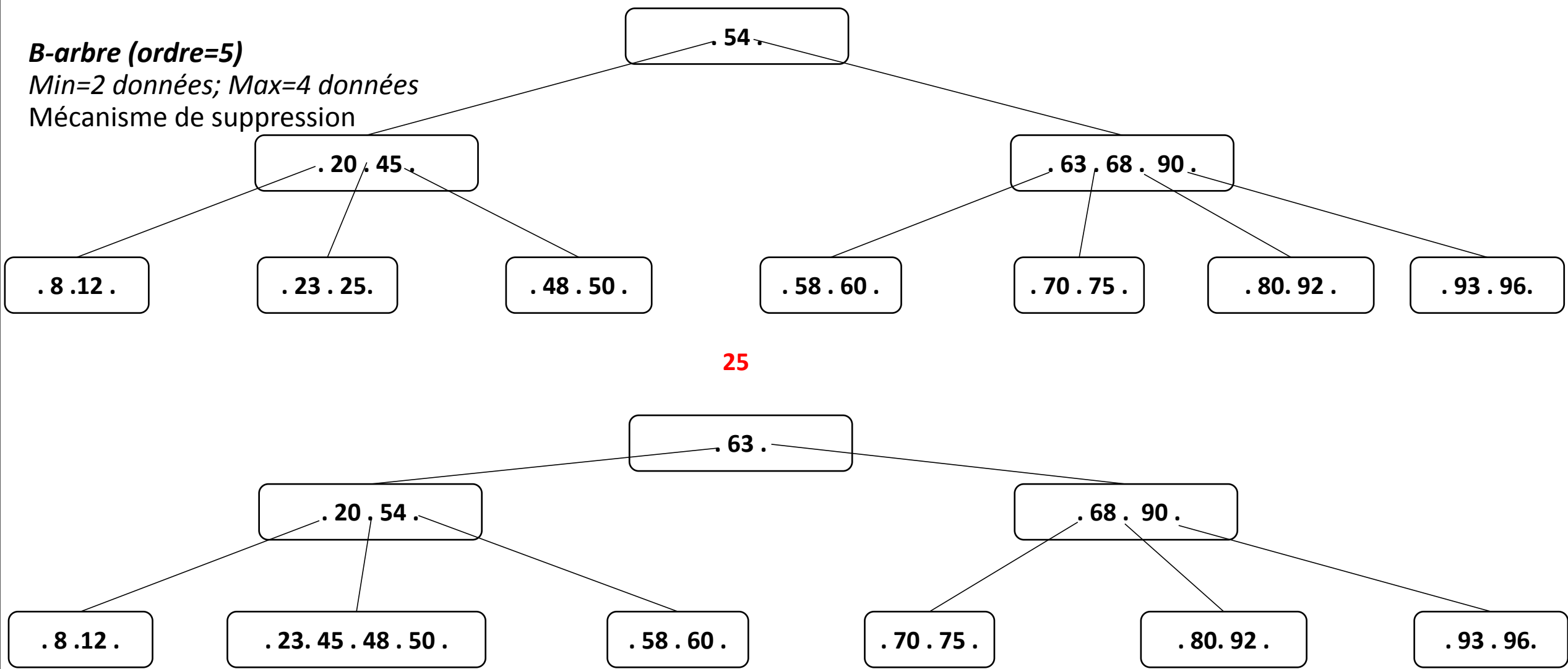
60



Les arbres m-aires

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.

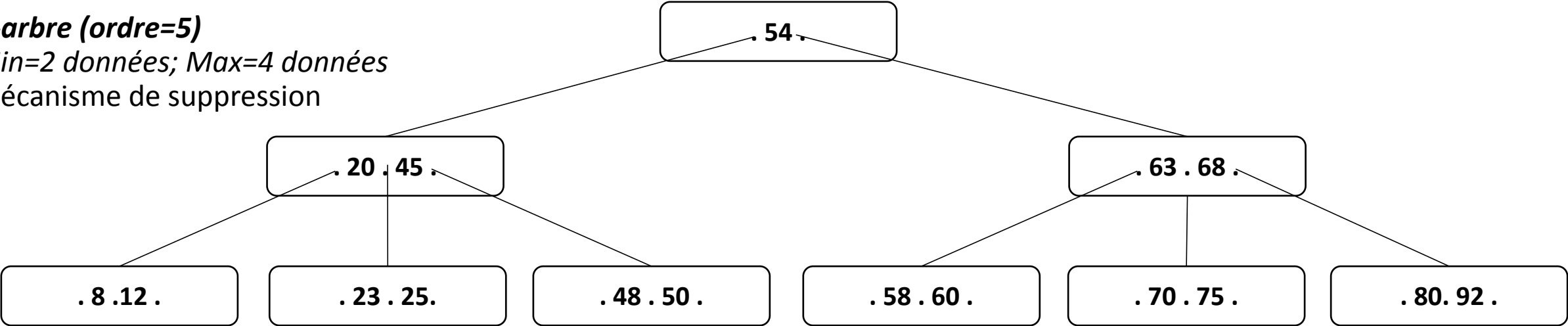
B-arbre (ordre=5)
Min=2 données; Max=4 données
Mécanisme de suppression



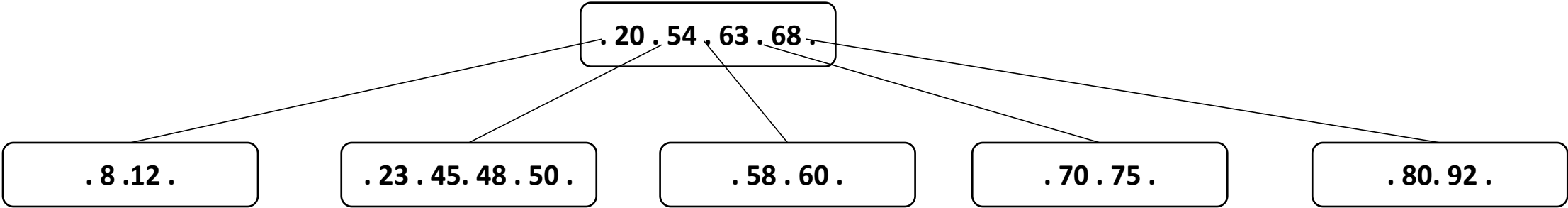
Les arbres m-aires

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

B-arbre (ordre=5)
Min=2 données; Max=4 données
Mécanisme de suppression



23



Les arbres m-aires

Utilisation en RAM : Arbres 2-3 (Définition / Propriétés)

C'est un arbre de recherche m-aire équilibré (B-arbre) d'ordre 3

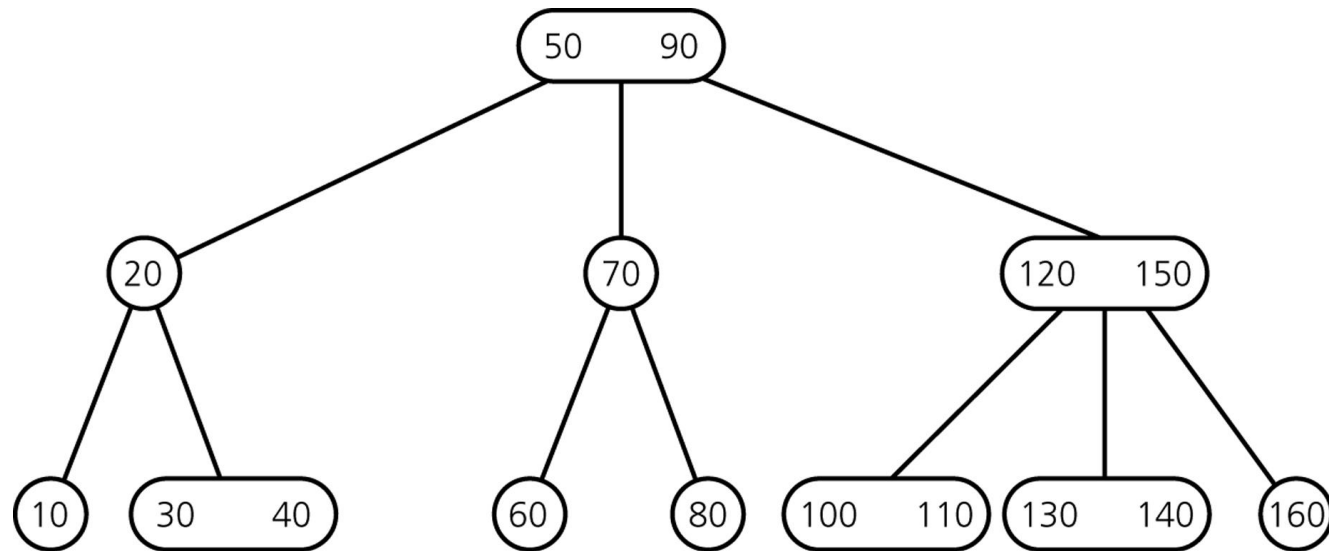
Equilibre garanti par construction

Nombre d'éléments dans un 2-3 tree de hauteur h est entre $2^h - 1$ et $3^h - 1$.

Donc, la hauteur d'un 2-3 tree avec n éléments est entre $\text{ENT}(\log_3 (N+1))$ et $\text{ENT}(\log_2 (N+1))$

Les arbres m-aires

Exemple d'un arbre 2-3



Les arbres m-aires

Utilisation en RAM : Arbres 2-4 (Définition / Propriétés)

C'est un arbre de recherche m-aire équilibré (B-arbre) d'ordre 4

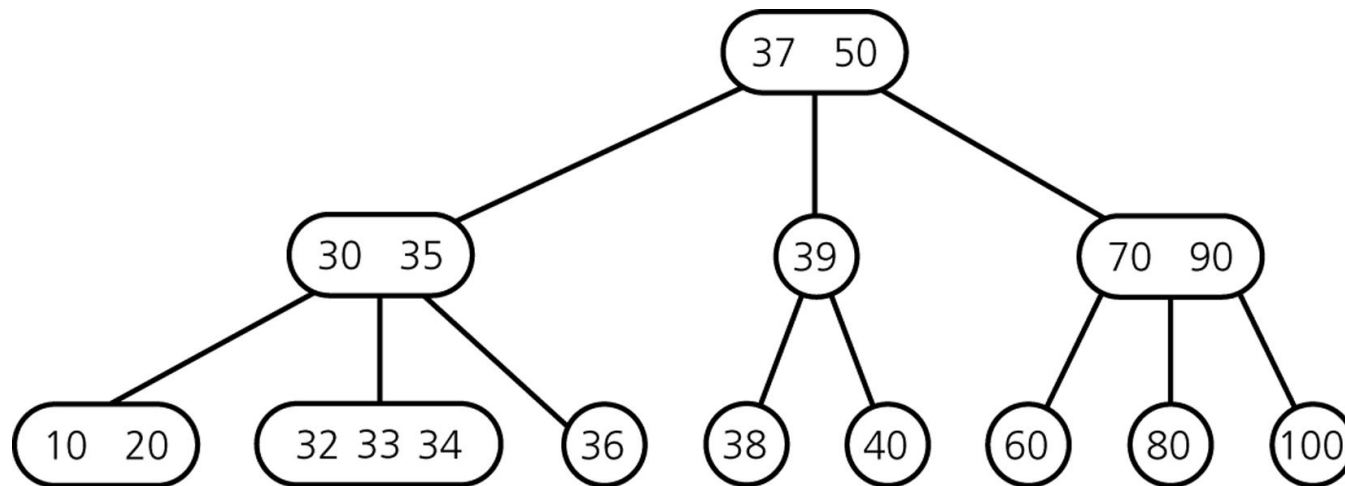
Equilibre garanti par construction

Nombre d'éléments dans un 2-4 tree de hauteur h est entre $2^h - 1$ et $4^h - 1$.

Donc, la hauteur d'un 2-4 tree avec n éléments est entre $\lceil \log_4 (N+1) \rceil$ et $\lceil \log_2 (N+1) \rceil$

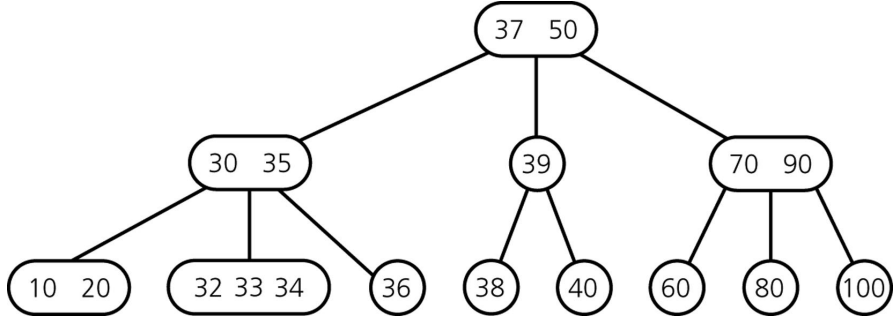
Les arbres m-aires

Exemple d'un arbre 2-4

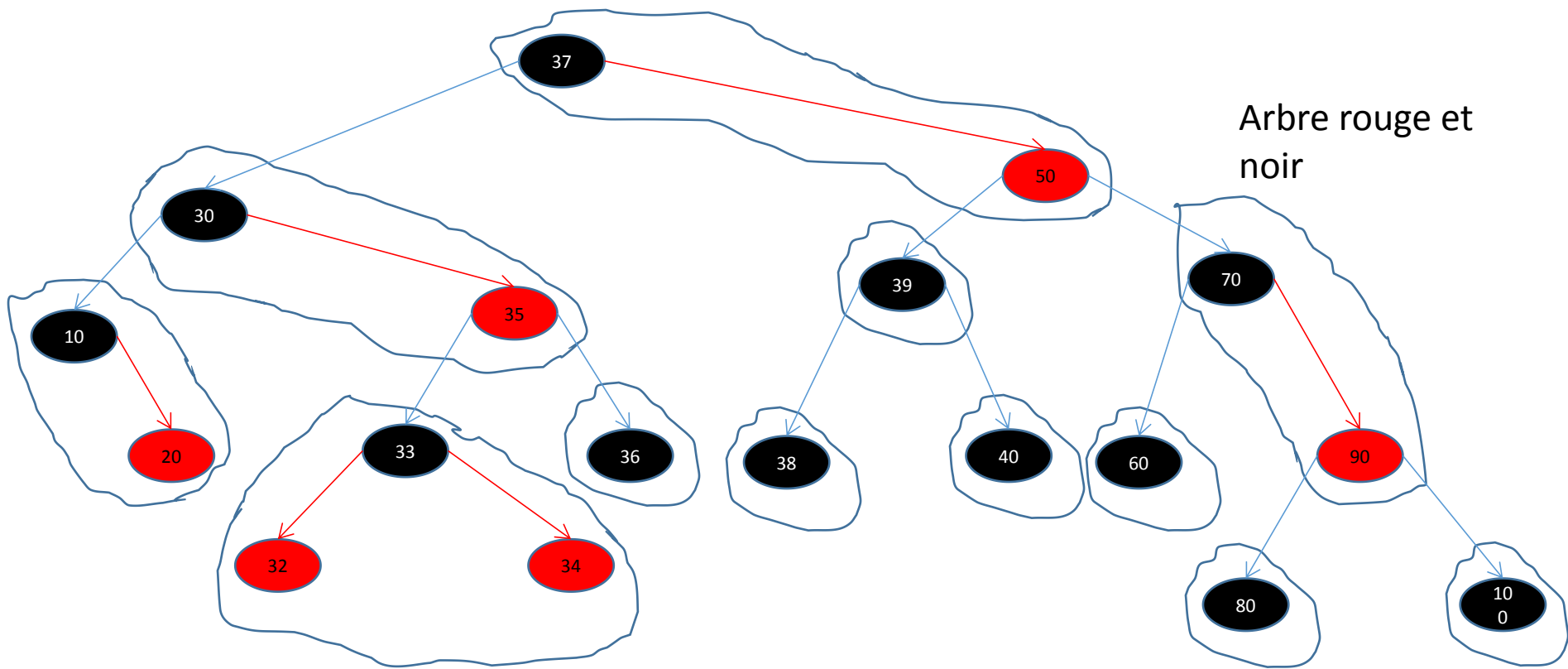


Les arbres m-aires

Arbres 2-4 vers Arbres RB



Arbre 2-4



Arbre rouge et noir

Les arbres m-aires

Implémentation : Arbre de recherche m-aire (Dynamique en C)

SOIT
A UN ARM (4) ;

DEBUT
FIN



```
/** Implémentation **\  
/** Arbres de recherche m-aire **/  
  
typedef int Typeelem_M4i ; // Type d'un élément  
typedef struct Noeud_M4i * Pointeur_M4i ; // Arbre  
  
struct Noeud_M4i  
{  
    int Degre ;  
    Pointeur_M4i Fils[4] ;  
    Typeelem_M4i Infor[4] ;  
    Pointeur_M4i Parent ;  
};
```

Les arbres m-aires

Implémentation : Arbre de recherche m-aire (Dynamique en C)

```
void Creernoed_M4i( Pointeur_M4i *P )
{
    int I ;

    *P = (struct Noeud_M4i *) malloc( sizeof( struct Noeud_M4i)) ;
    for (I=0; I< 4; ++I) (*P)->Fils[I] = NULL;
    (*P)->Degre = 0 ;
    (*P)->Parent = 0 ;
}

void Liberernoed_M4i(Pointeur_M4i P)
{ free ( P );}
```

Les arbres m-aires

Implémentation : Arbre de recherche m-aire (Dynamique en C)

```
Typeelem_M4i Infor_M4i(Pointeur_M4i P, int I)  
{ return P->Infor[I-1] ; }
```

```
Pointeur_M4i Fils_M4i( Pointeur_M4i P, int I)  
{ return P->Fils[I-1] ; }
```

```
Pointeur_M4i Parent_M4i( Pointeur_M4i P)  
{ return P->Parent ; }
```

```
int Degre_M4i ( Pointeur_M4i P )  
{ return P->Degre ; }
```

```
void Aff_infor_M4i ( Pointeur_M4i P, int I, Typeelem_M4i Val)  
{  
    P->Infor[I-1] = Val ;  
}
```

```
void Aff_fils_M4i( Pointeur_M4i P, int I, Pointeur_M4i Q)  
{ P->Fils[I-1] = Q ; }
```

```
void Aff_parent_M4i( Pointeur_M4i P, Pointeur_M4i Q)  
{ P->Parent = Q ; }
```

```
void Aff_degre_M4i ( Pointeur_M4i P, int N)  
{ P->Degre = N ; }
```

Les arbres m-aires

Implémentation : Arbre m-aire (Dynamique en C)

```
/** Implémentation **\  
/** Arbre m-aire **/  
  
typedef int Typeelem_M4i ; // Type d'un élément  
typedef struct Noeud_M4i * Pointeur_M4i ; // Arbre  
  
struct Noeud_M4i  
{  
    int Degre ;  
    Pointeur_M4i Fils[4] ;  
    Typeelem_M4i Infor;  
    Pointeur_M4i Parent ;  
};
```

Les arbres m-aires

Implémentation : Statique

Arbre m-aire

Représenté par un tableau

Chaque élément du tableau contient les champs:

- Information
- Degré
- Tableau d'indice vers les fils
- Père (Optionnel)
- Bit d'effacement

Arbre de recherche m-aire

Représenté par un tableau

Chaque élément du tableau contient les champs:

- Tableau des Informations
- Tableau d'indices vers les fils
- Degré
- Père (Optionnel)
- Un tableau de bits d'effacement