

IFT339

Structures de données

Thème 11 : Arbres balancés, Arbres AA Arbres rouges et noirs

Aïda Ouangraoua
Département d'informatique



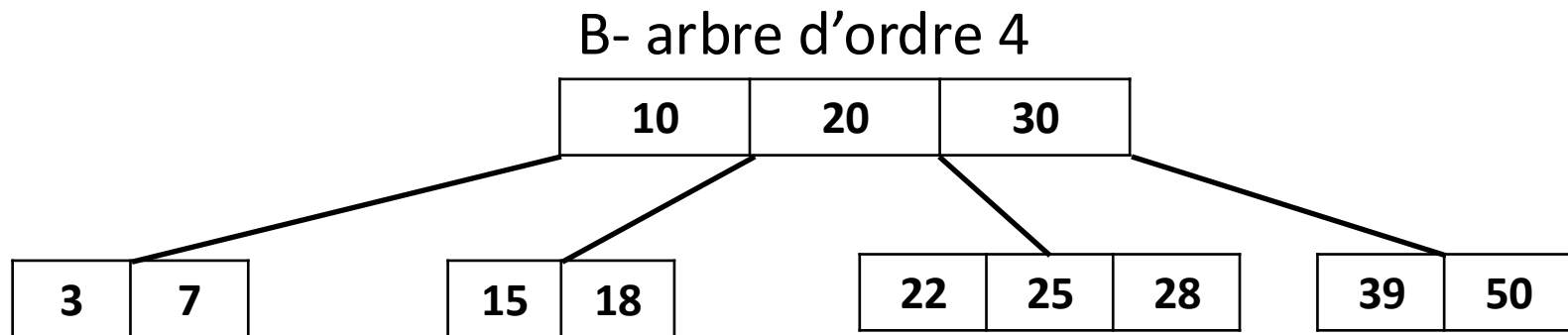
UNIVERSITÉ DE
SHERBROOKE

Arbres balancés (B-arbre)

- ☐ Avec AVL : recherche en $O(\log(n))$
- ☐ Comment faire mieux pour des ensembles très grands (n très grand)
- ☐ Solution : relâcher la contrainte d'arbre binaire pour des avoir des arbres plus larges et de faible hauteur
- ☐ Contrainte: toutes les feuilles doivent avoir la même hauteur

Arbres balancés (B-arbre) d'ordre m

- ❑ Chaque nœud a entre $m/2$ et m enfants (la racine, entre 2 et m enfants)
 - ❑ Un nœud qui a k enfants contient $k-1$ éléments
 - ❑ Entre chaque paire d'éléments consécutifs, il y a un enfant (sous-arbre)



- ❑ Nombre min/max d'éléments dans un B-arbre d'ordre 100 et de hauteur 3
 - max = $99 + 100 \cdot 99 + 100 \cdot 100 \cdot 99$
 - min = $1 + 2 \cdot 49 + 2 \cdot 50 \cdot 49$
- ❑ Pour un AVL de hauteur 3 : max = 7 ; min = 3

Insertion de x dans un B-arbre

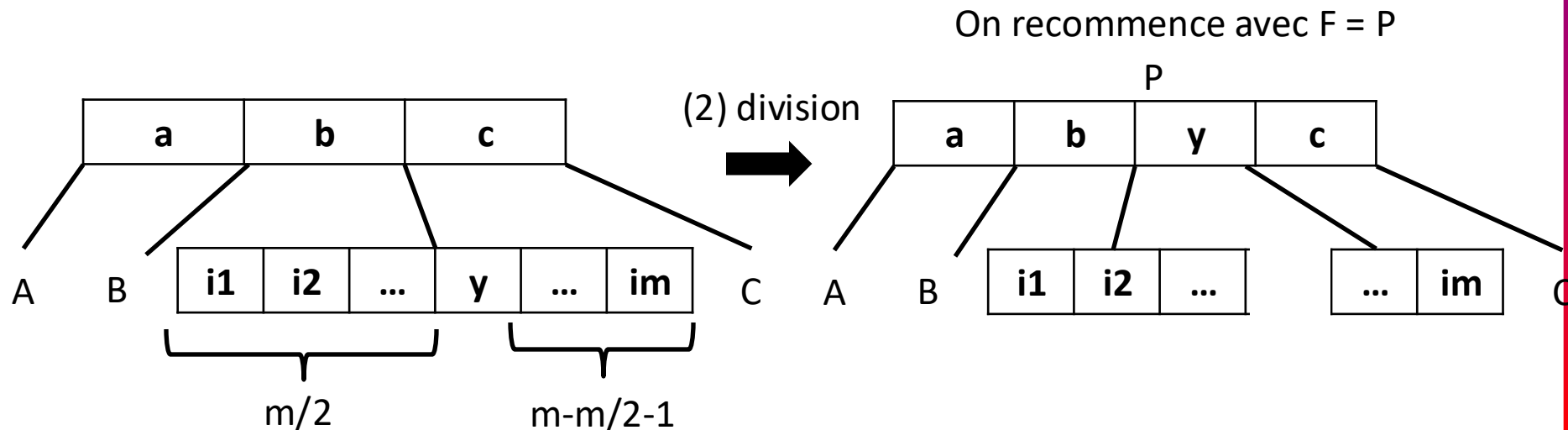
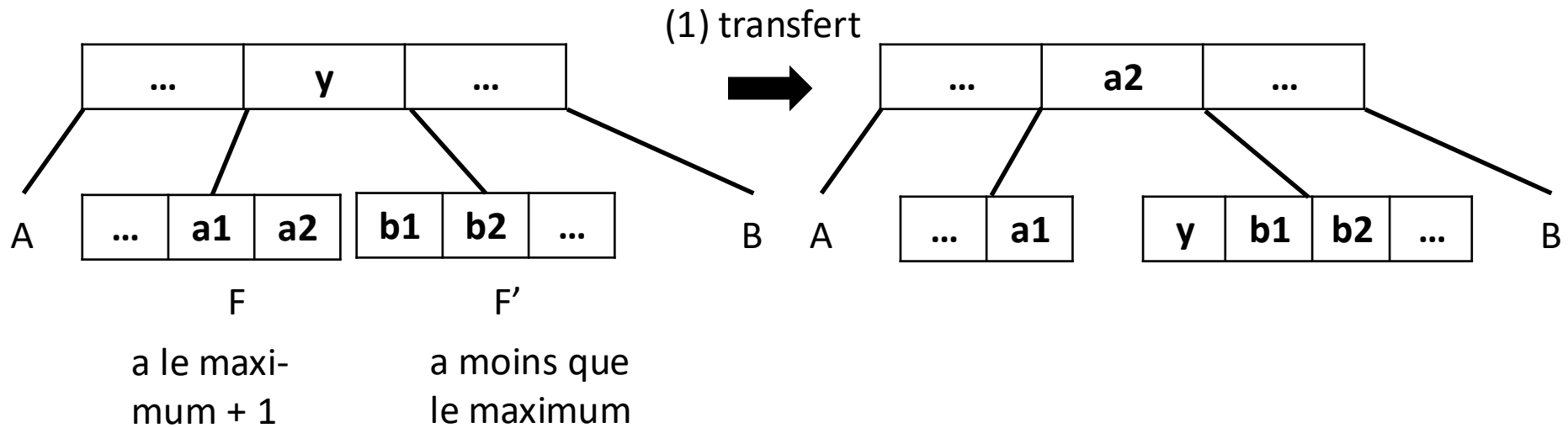
- ❑ Localiser la feuille F où insérer
- ❑ Insérer x dans F
- ❑ Tant que F est saturé, i.e. F a plus de $m-1$ éléments:

On insère toujours dans une feuille

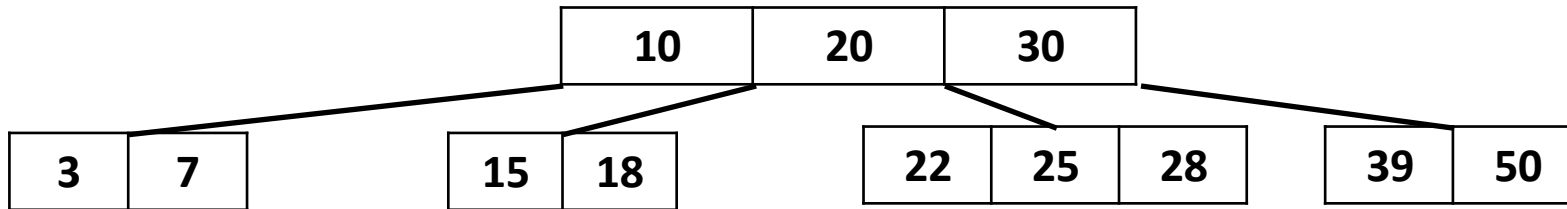
Facultatif

- ❑ (1-Transfert) Si F a un frère adjacent F' qui a moins que $m-1$ éléments
 - ❑ Transférer un élément de F vers le parent et un élément du parent vers F'
- ❑ (2-Division) Sinon:
 - ❑ Diviser F en deux feuilles F1 et F2 telles que F1 contient les $m/2$ premiers éléments F2 les $m - m/2 - 1$ derniers et y est l'élément du milieu
 - ❑ Insérer y dans le parent P de F avec F1 comme enfant de P à gauche de y, et F2 comme enfant de P à droite de y
 - ❑ $F = P$

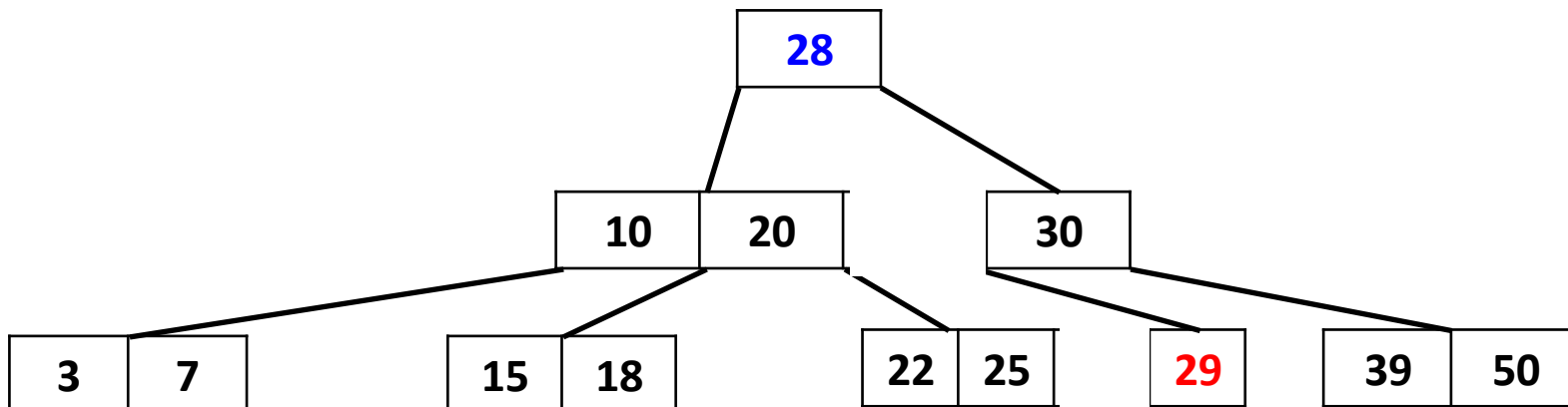
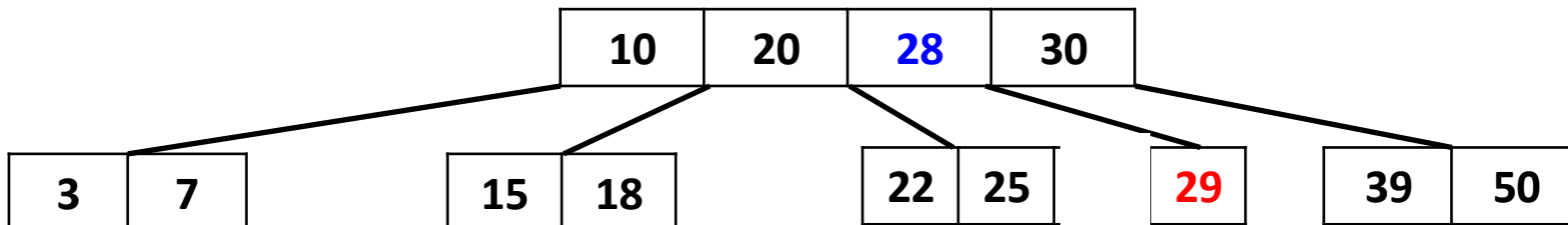
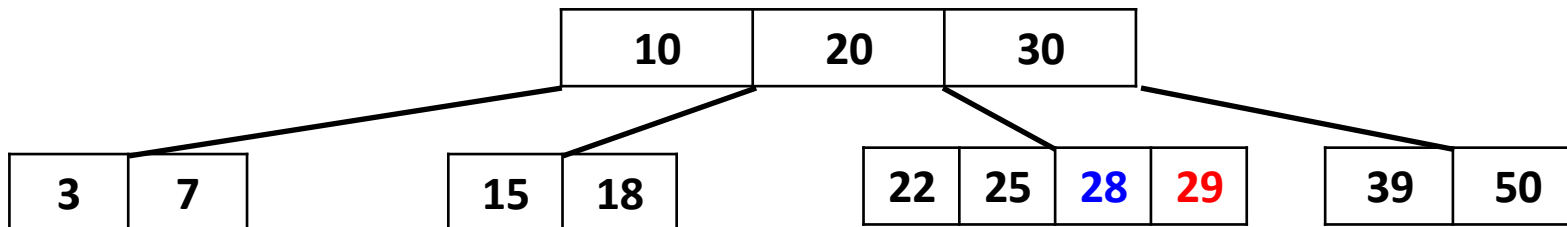
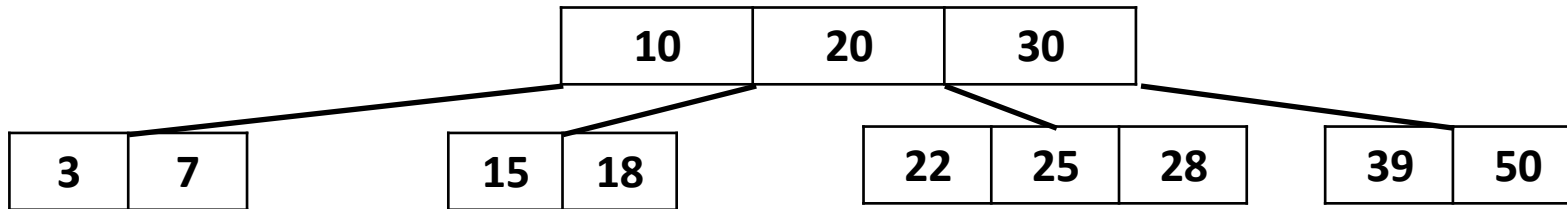
Insertion de x dans un B-arbre



Exemple: Insertion de 29 (ordre 4)



Exemple: Insertion de 29 (ordre 4)



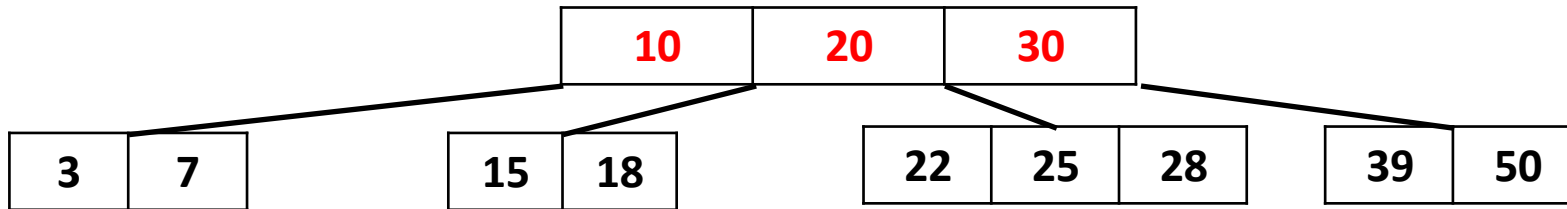
Autre algorithme d'insertion pour $m > 2$: la taupe

- ❑ Ne jamais descendre dans un nœud plein:
- ❑ À partir de la racine, descendre vers la feuille F où insérer x
- ❑ Pour chaque nœud N rencontré:
 - ❑ Si N est plein, i.e N a $m-1$ éléments:
 - ❑ Diviser N en en deux feuilles N1 et N2 telles que
N1 contient les $(m-1)/2$ premiers éléments
F2 les $(m-1) - (m-1)/2 - 1$ derniers
et y est l'élément du milieu
 - ❑ Insérer y dans le parent de N
- ❑ Insérer x dans le dernier nœud (feuille) F

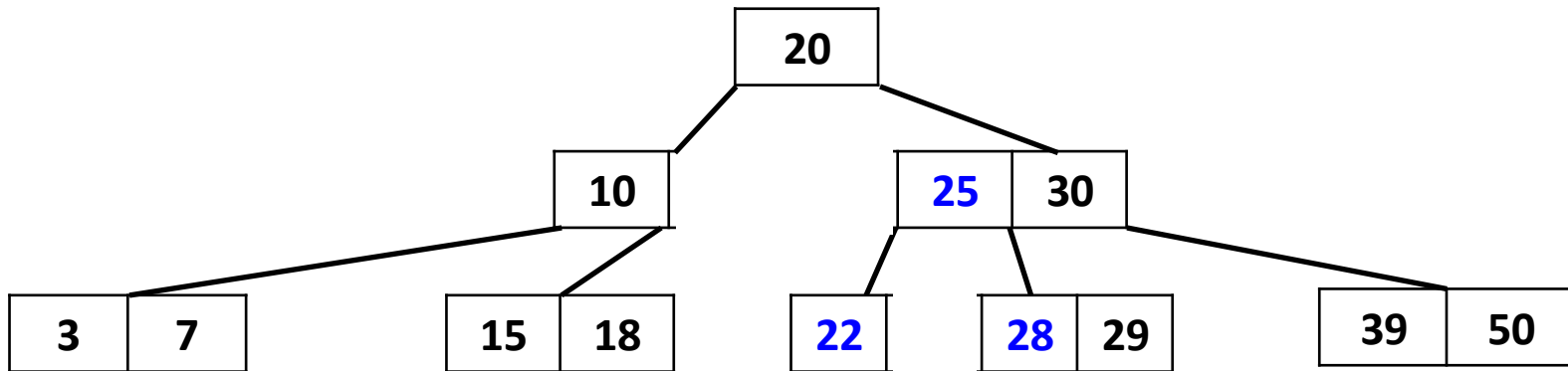
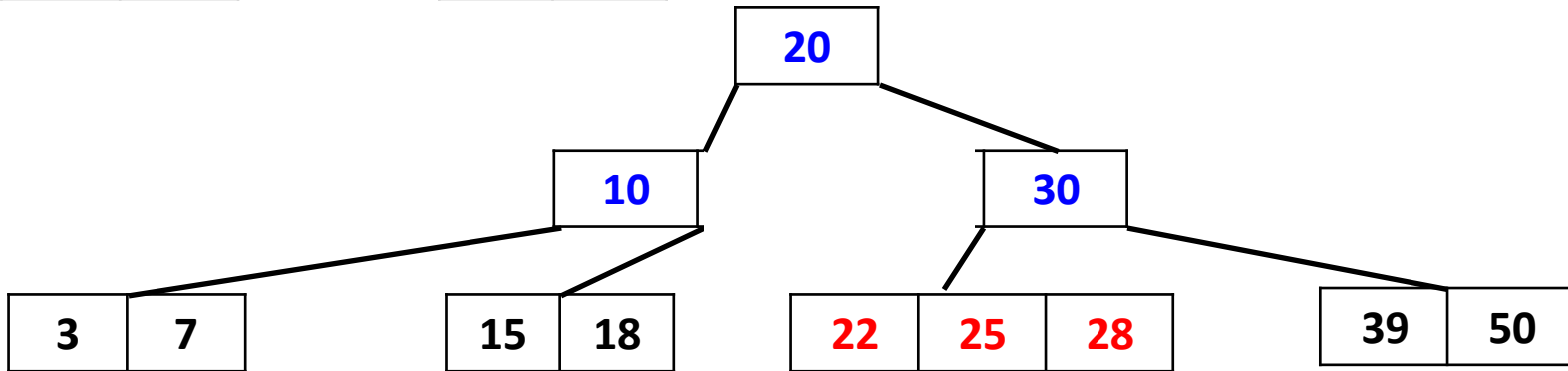
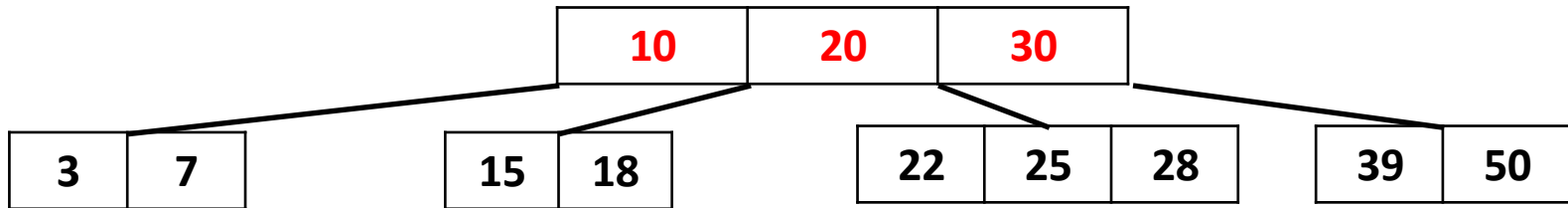
Aucun nœud sur le chemin de la racine à F ne peut être saturé, puisque tous les nœuds plein sur le chemin on été divisés.

On insère toujours dans une feuille

Exemple: Insertion de 29 (ordre 4)



Exemple: Insertion de 29 (ordre 4)

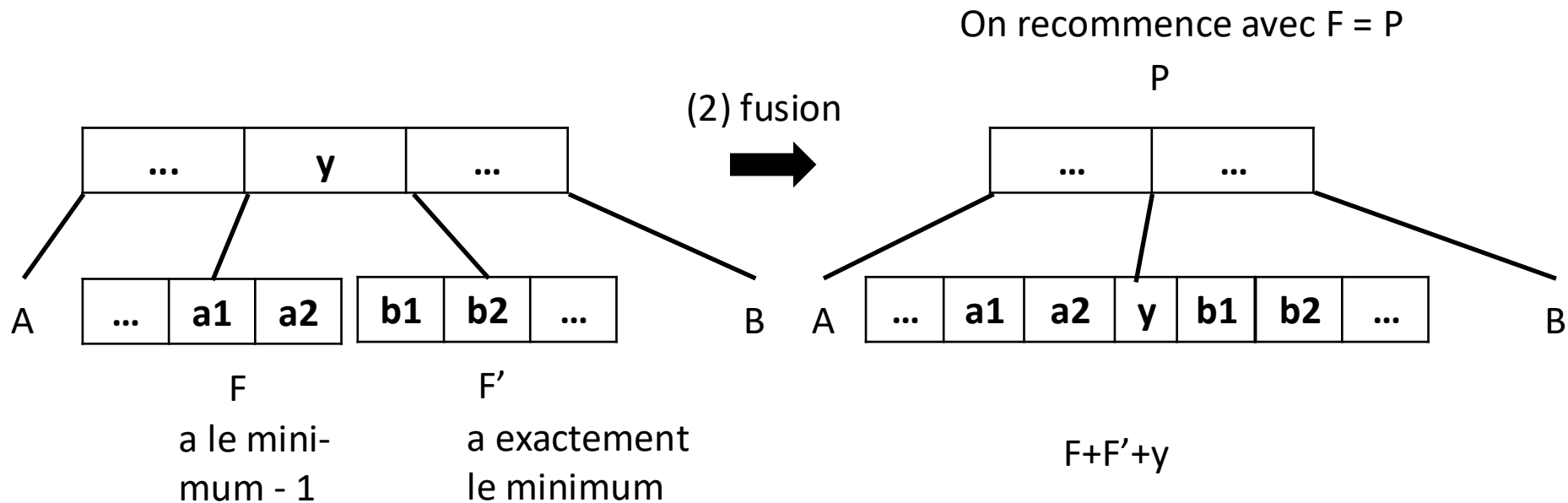
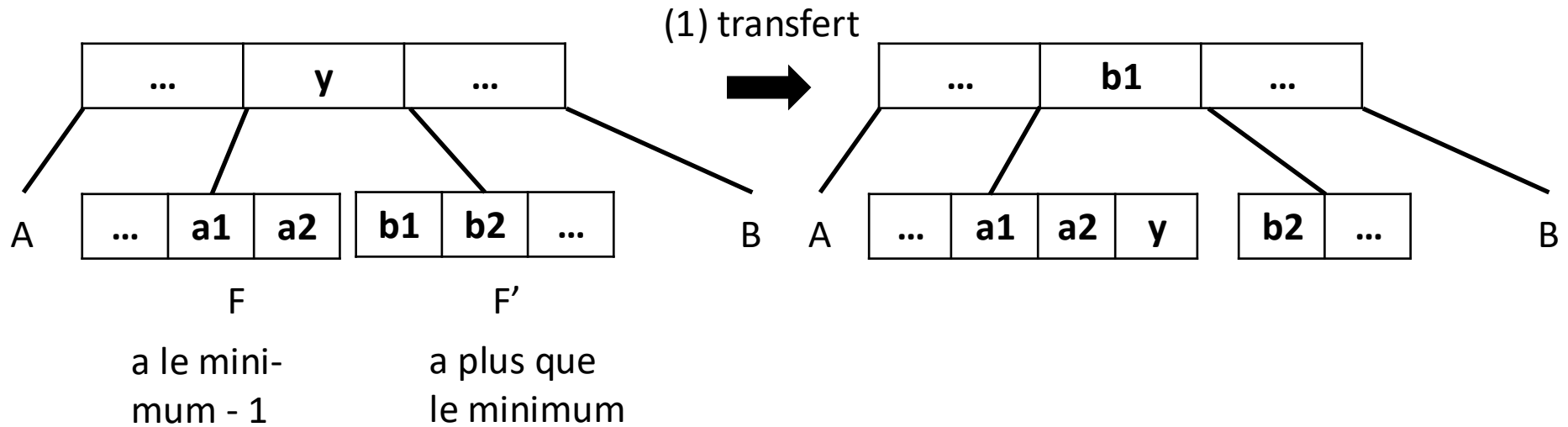


Suppression de x dans un B-arbre

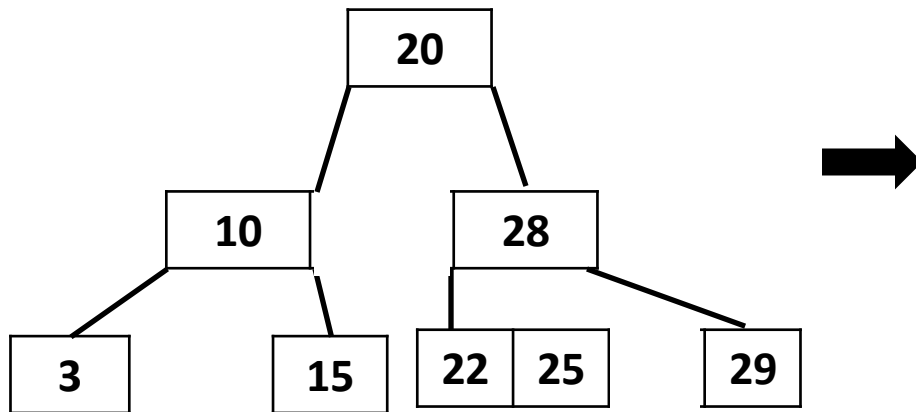
- ☐ Localiser le nœud N où se trouve x
- ☐ Si N est une feuille
 - ☐ $F = N$; Supprimer x de F
- ☐ Sinon
 - ☐ localiser la feuille F contenant le précédent x' de x
 - ☐ Échanger les valeurs de x et x'
 - ☐ Supprimer x de F
- ☐ Tant que F est sous-rempli, i.e. F a moins de $m/2 - 1$ éléments:
 - ☐ (1-Transfert) Si F a un frère adjacent F' qui a plus de $m/2 - 1$ éléments
 - ☐ Transférer un élément de F' vers le parent et un élément du parent vers F
 - ☐ (2-Fusion) Sinon:
 - ☐ Fusionner de F avec un un frère adjacent F' + un élément du parent P
 - ☐ $F = P$

On supprime toujours dans une feuille

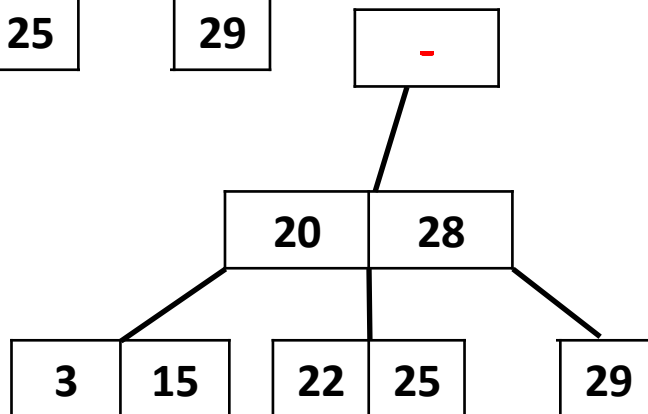
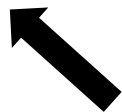
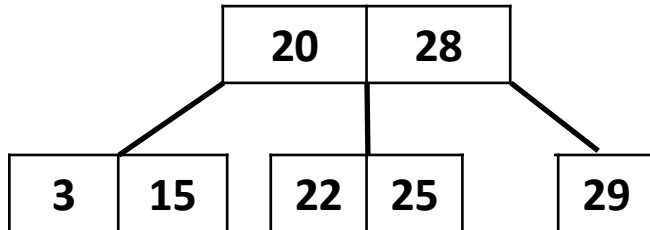
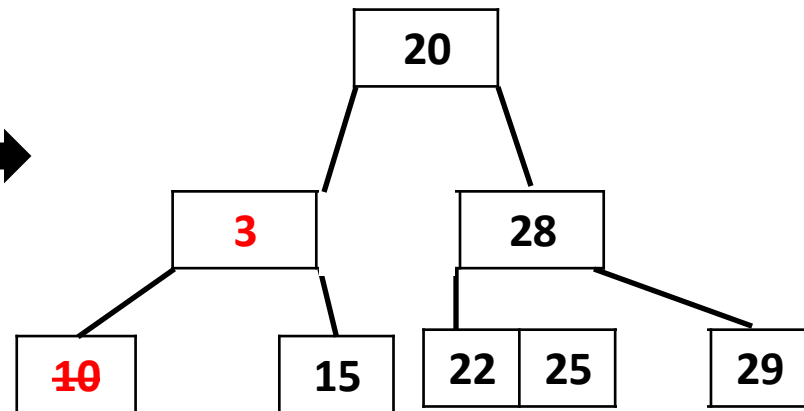
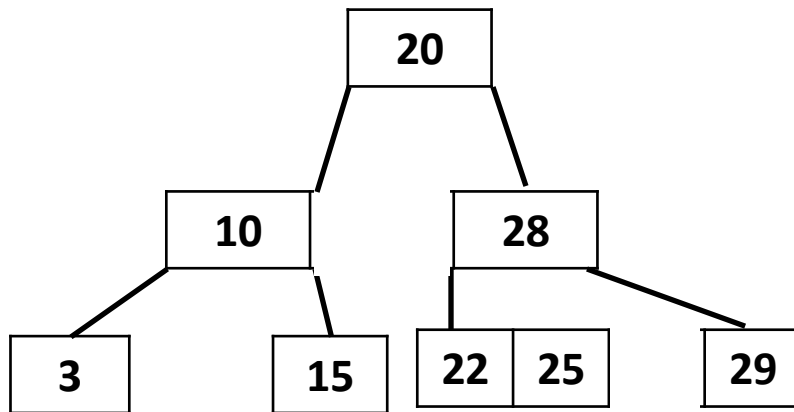
Suppression de x dans un B-arbre



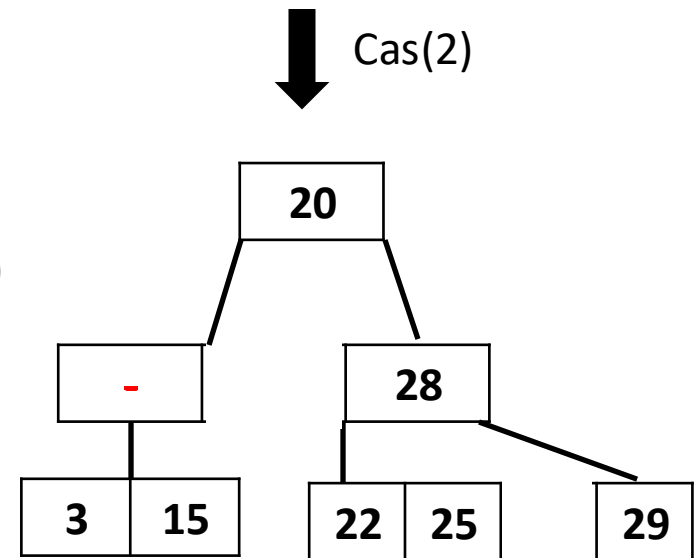
Exemple: Suppression de 10 (ordre 3)



Exemple: Suppresion de 10 (ordre 3)

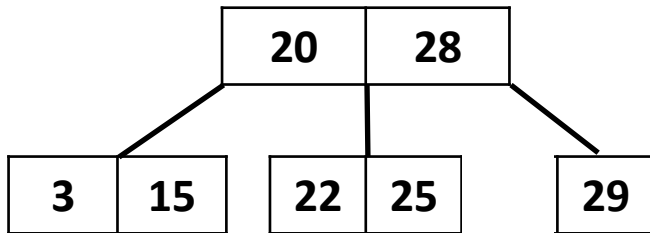


Cas(2)

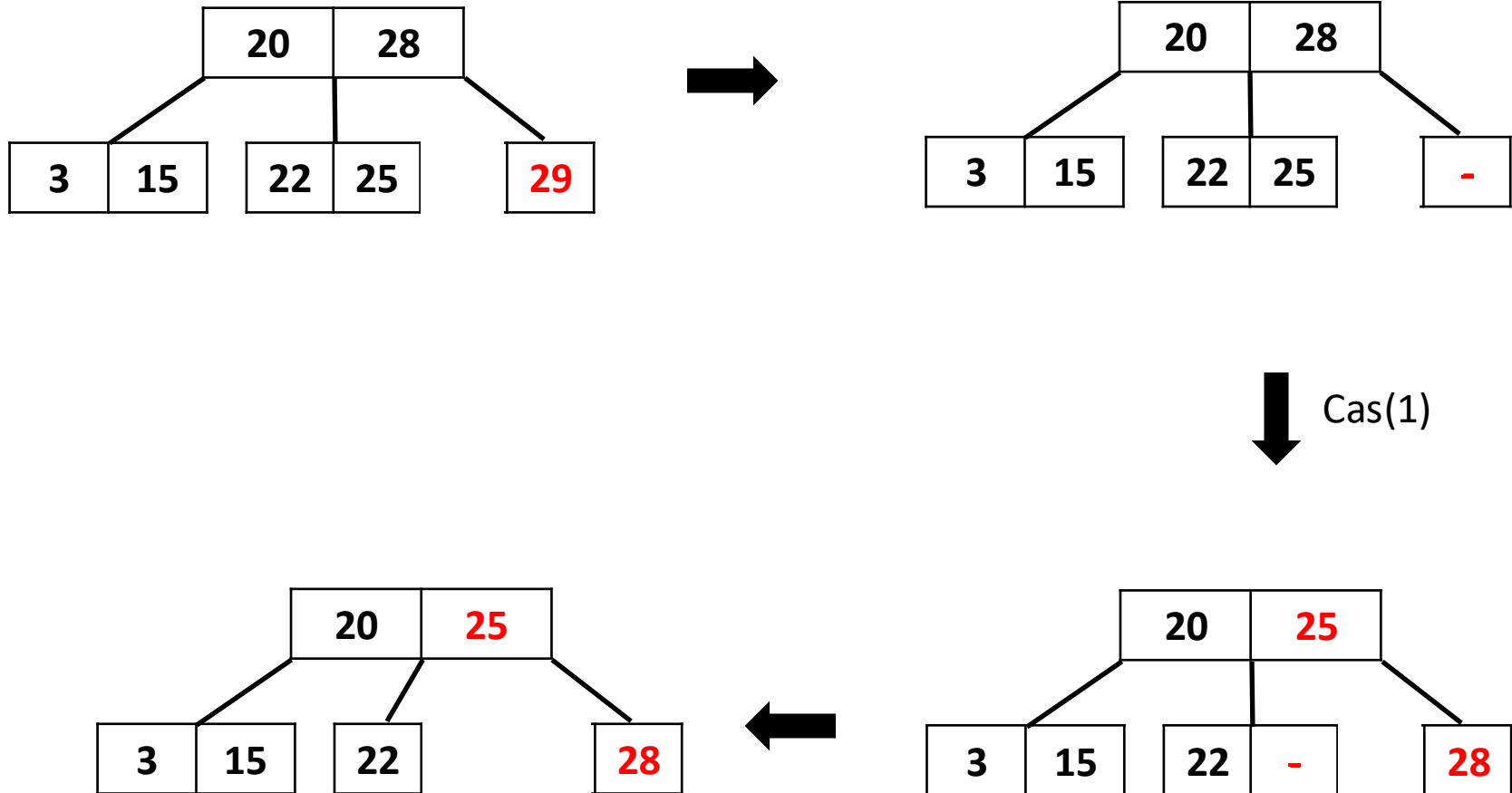


Cas(2)

Exemple: Suppresion de 29 (ordre 3)

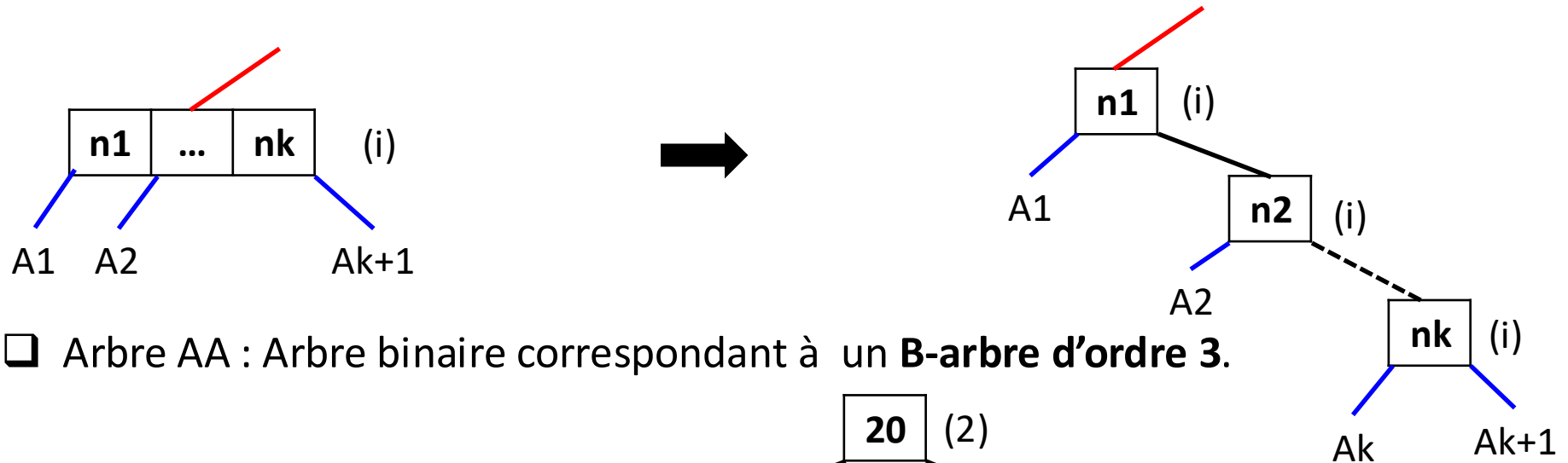


Exemple: Suppression de 29 (ordre 3)

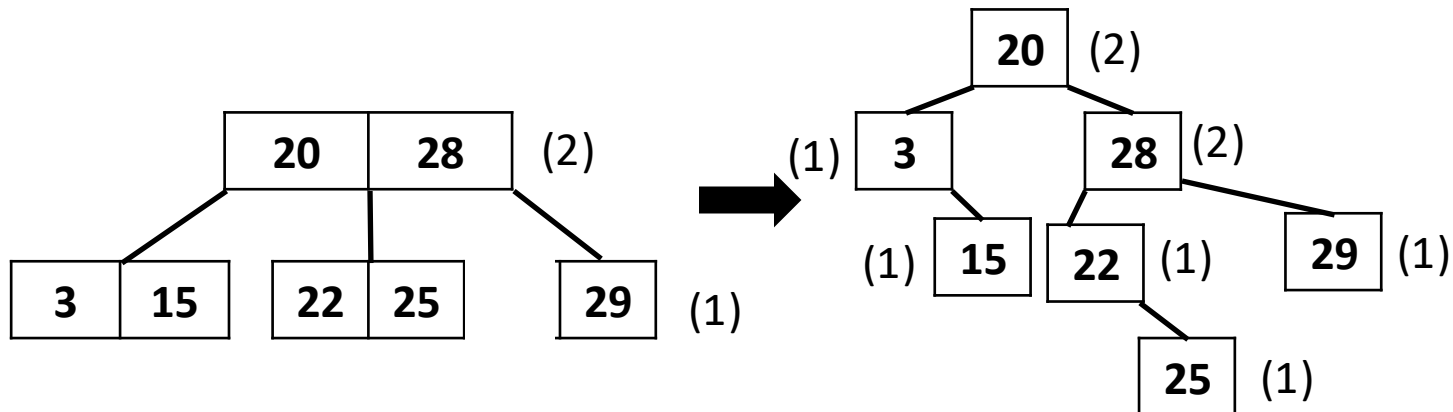


Arbres AA

- ❑ B-arbre donne une meilleure complexité pour la recherche que les arbres AVL, mais la taille variable des nœuds (entre $m/2 - 1$ et $m-1$) est difficile à manipuler
- ❑ Solution tout B-arbre peut être représenté en arbre binaire
 - ❑ Scinder chaque nœud en autant de niveaux qu'il a d'éléments
 - ❑ Associer à chaque élément son niveau initial (1 pour une feuille)

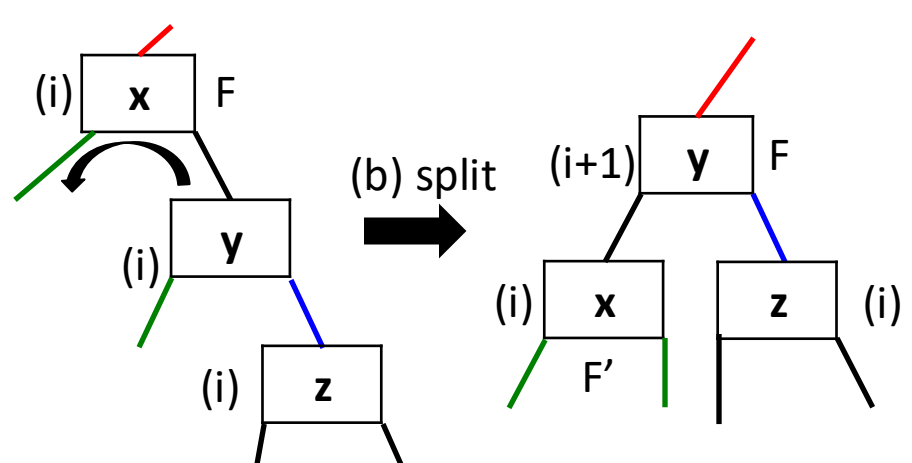
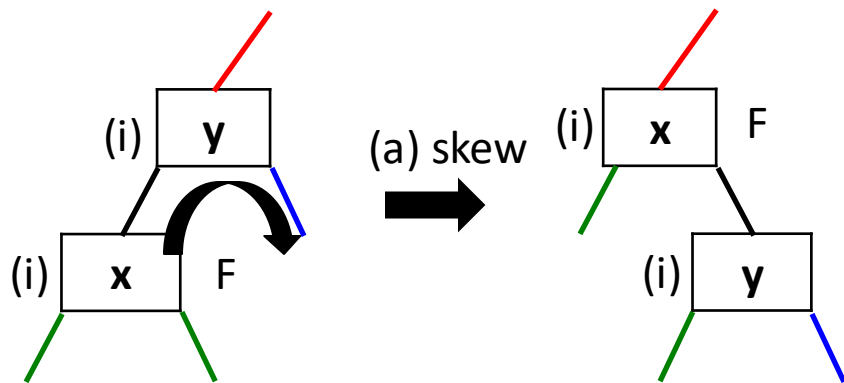


- ❑ Arbre AA : Arbre binaire correspondant à un **B-arbre d'ordre 3**.

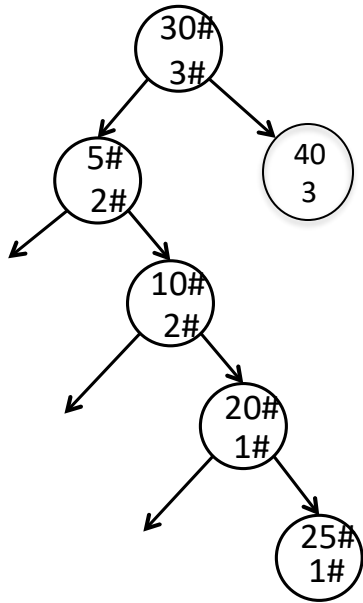


Insertion de x dans un AA-arbre

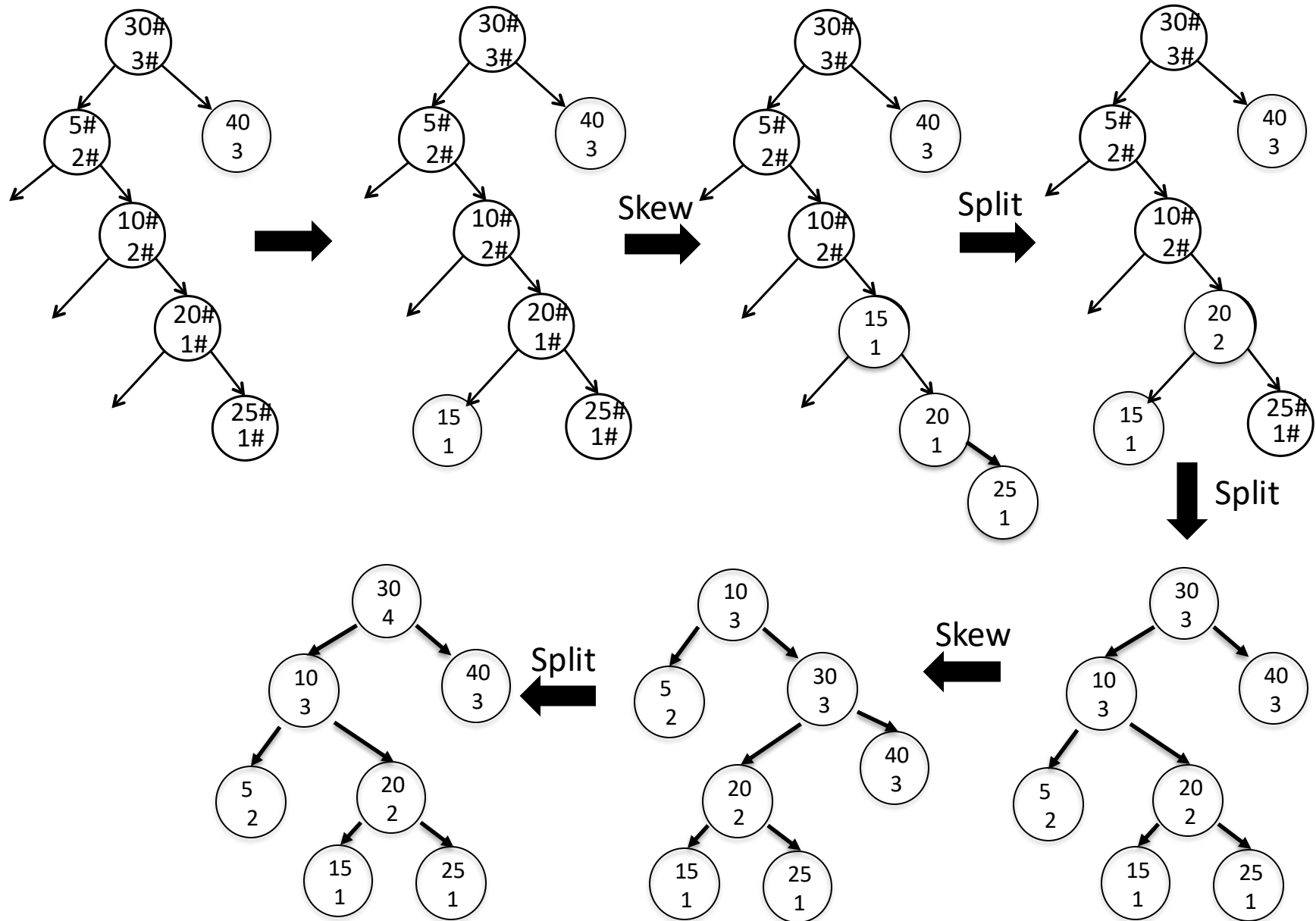
- ❑ Localiser là où insérer x
- ❑ Insérer x dans une nouvelle feuille F de niveau (1)
- ❑ niveauPrec = 0
- ❑ Tant que niveau(F) != niveauPrec
 - ❑ Si F est enfant gauche de son parent
 - ❑ (a) Faire un **skew** (rotation droite) pour aligner F à droite
 - ❑ Si F contribue à un ensemble E saturé (i.e. contient plus de 2 éléments)
 - ❑ (b) Faire un **split** pour faire monter le milieu de E
 - ❑ Faire une rotation gauche pour ramener le milieu au centre
 - ❑ F = milieu de E ;
 - ❑ niveau(F)++
- ❑ niveauPrec++



Exemple: Insertion de 15 dans le AA-arbre

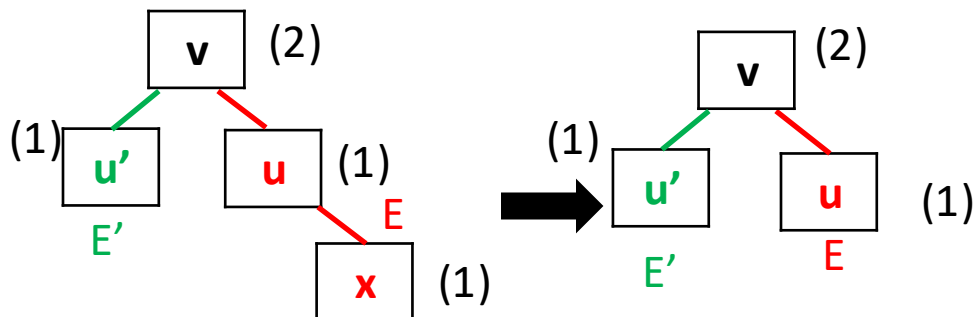


Exemple: Insertion de 15 dans le AA-arbre

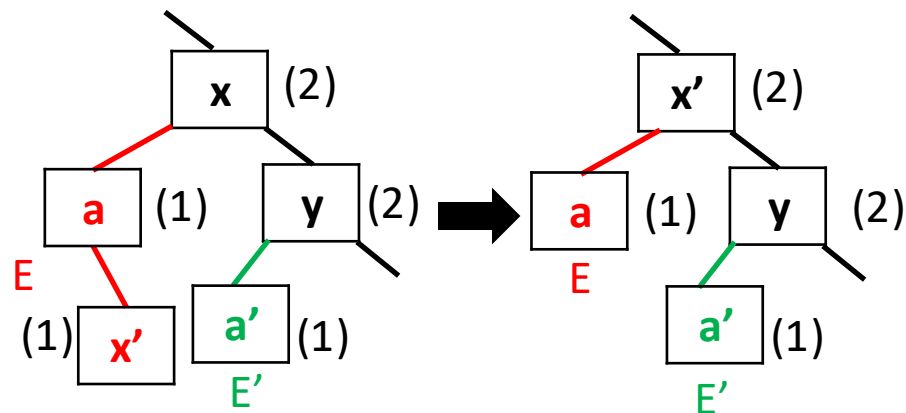


Suppression de x dans un AA-arbre

- ❑ Localiser le nœud N où se trouve x
- ❑ Si N est une feuille
 - ❑ E = ensemble auquel contribue N
 - ❑ Supprimer N



- ❑ Sinon
 - ❑ localiser la feuille N contenant le précédent ou le suivant x' de x
 - ❑ Échanger x et x'
 - ❑ E = ensemble auquel contribue N
 - ❑ Supprimer N

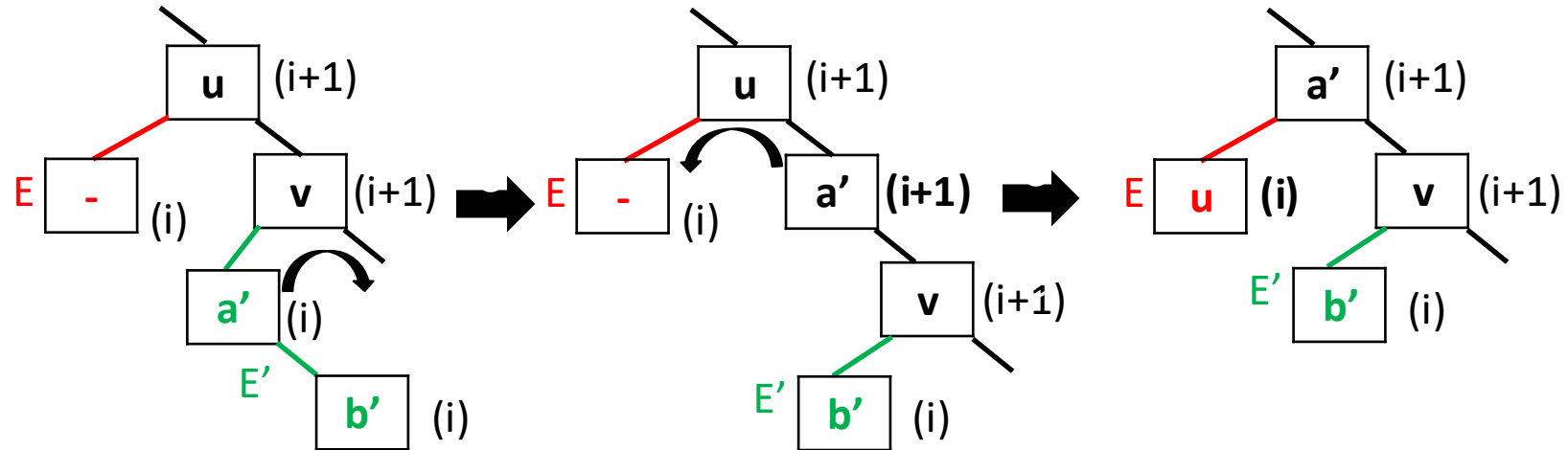


- ❑ Tant que E est un ensemble sous-rempli (i.e. contient 0 éléments):
 - ❑ (a) (Transfert) Si le frère adjacent E' de E a plus de 1 élément (i.e 2 éléments)
 - ❑ Transférer un élément un élément du parent vers E et de E' vers l'ensemble parent (rotation + incrémentations et décréments de niveaux)
 - ❑ (b) (Fusion) Sinon (i.e E' a 1 élément):
 - ❑ Fusionner de E avec E' + un élément du parent P ((rotations + décréments de niveau))
 - ❑ $E = P$

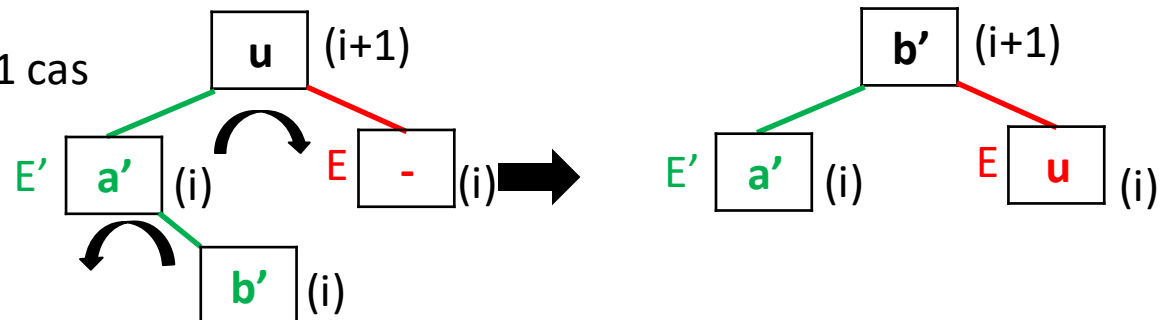
Suppression de x dans un AA-arbre

□ (a) Transfert

□ E à gauche de son parent : 2 cas



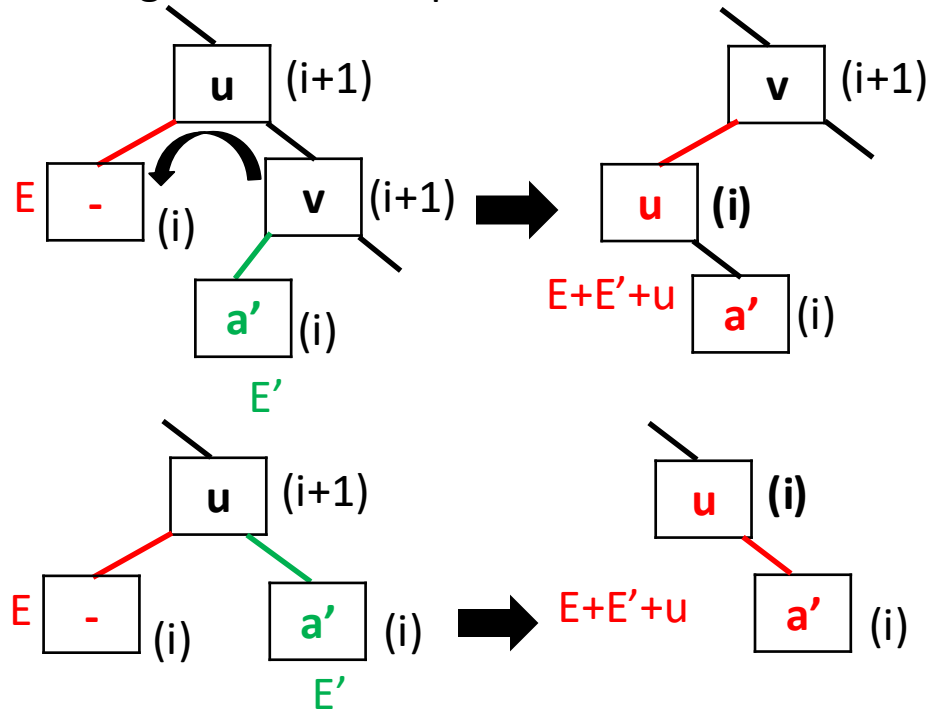
□ E à droite de son parent : 1 cas



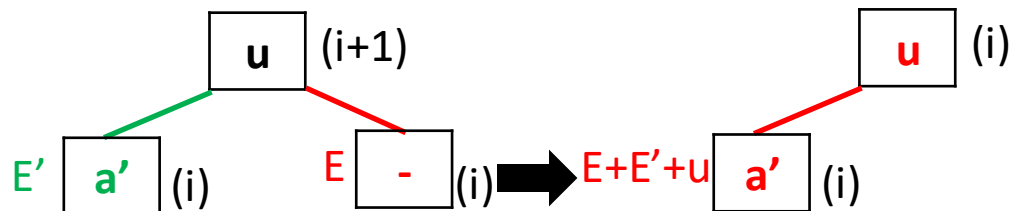
Suppression de x dans un AA-arbre

□ (b) Fusion

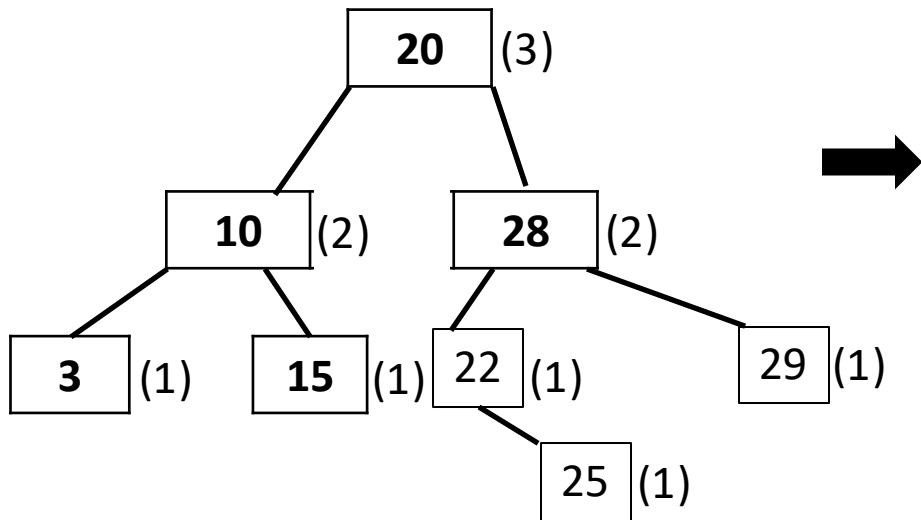
□ E à gauche de son parent : 2 cas



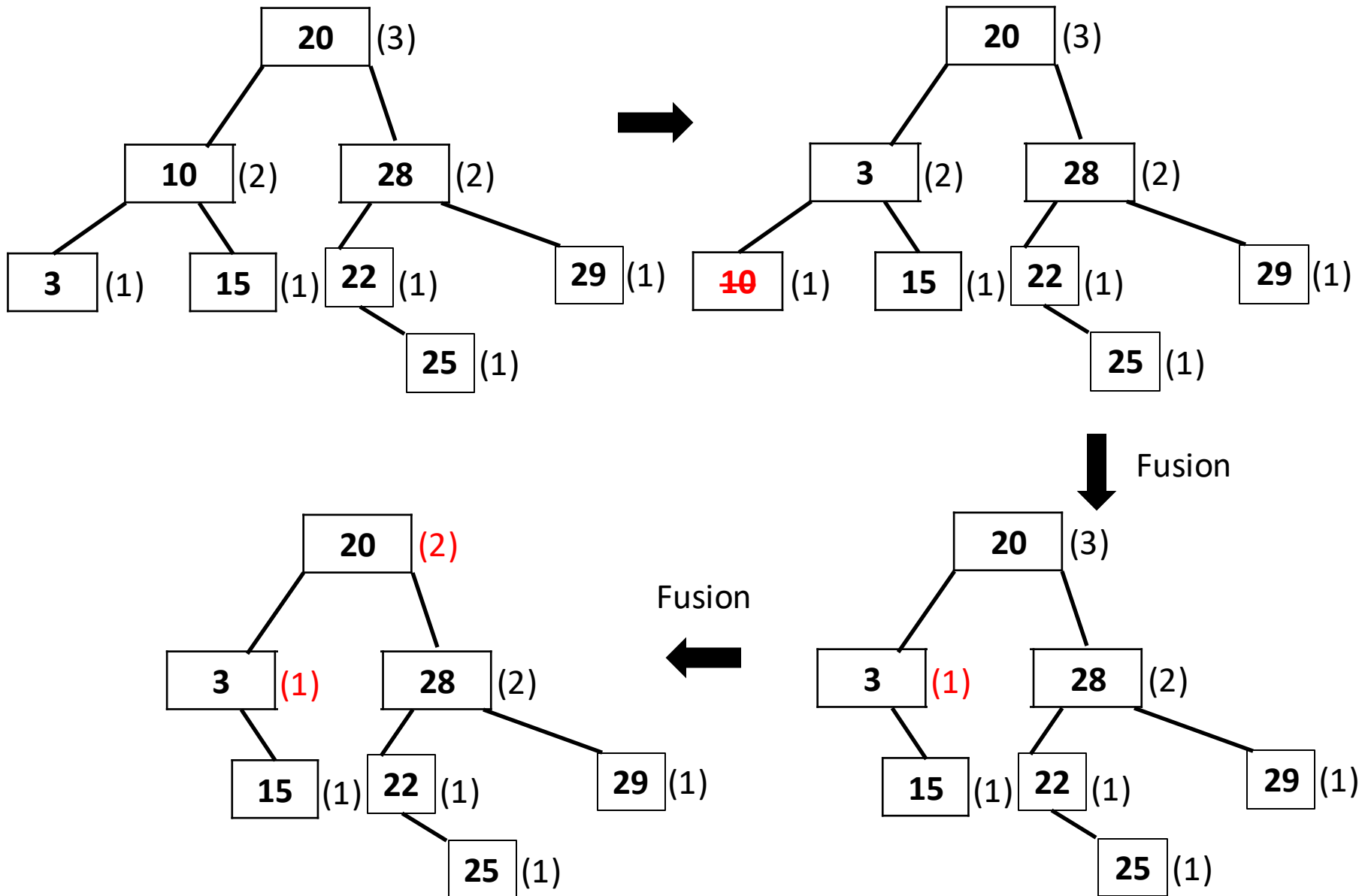
□ E à droite de son parent : 1 cas



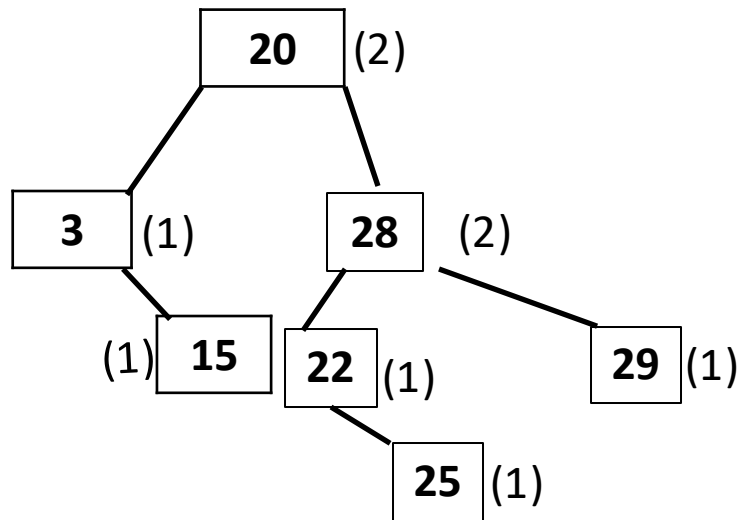
Exemple: Suppression de 10



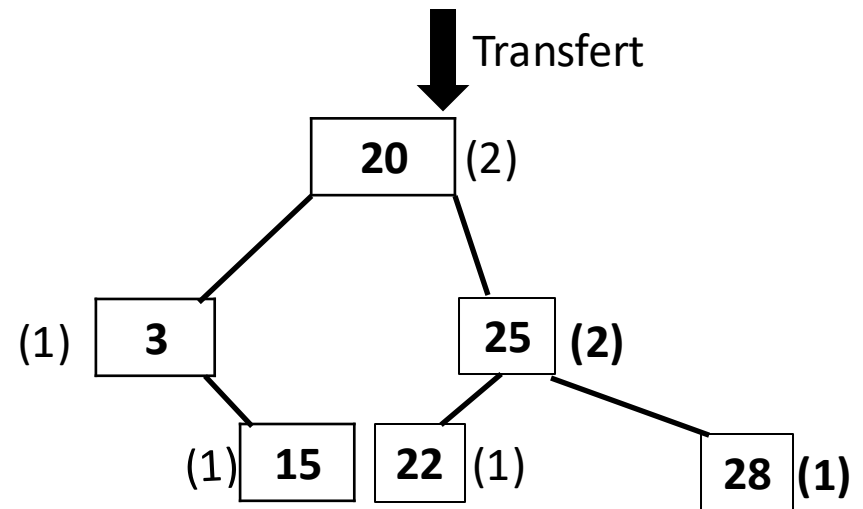
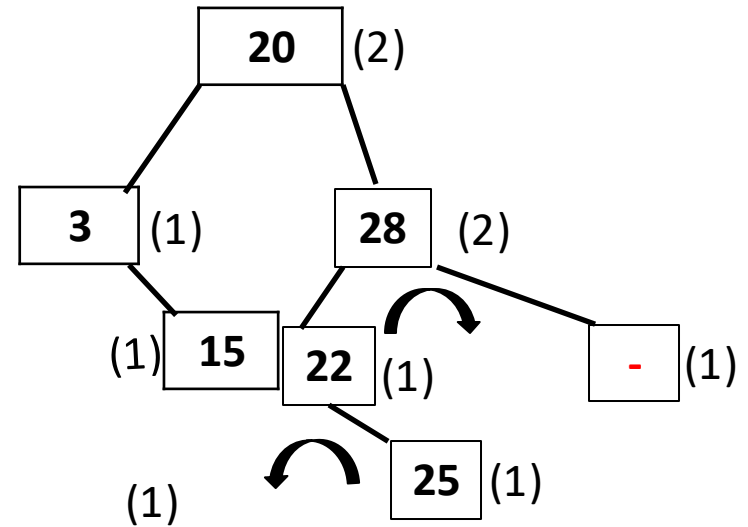
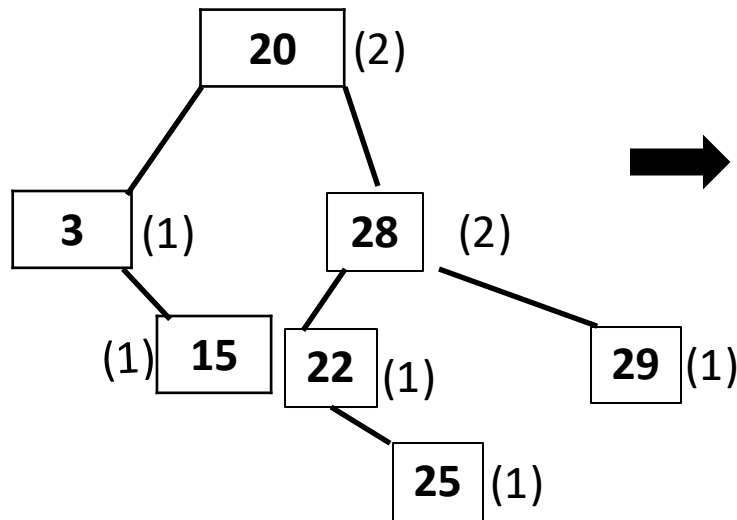
Exemple: Suppression de 10



Exemple: Suppression de 29

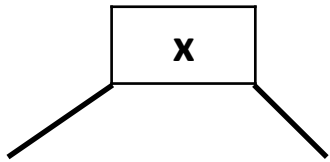


Exemple: Suppression de 29

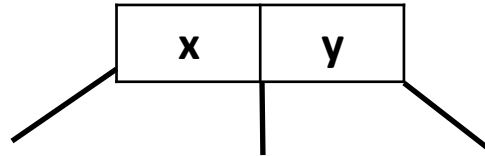
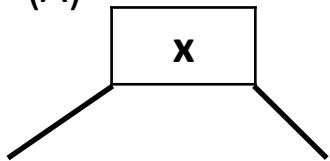


Arbres Rouge et Noir (bicolore)

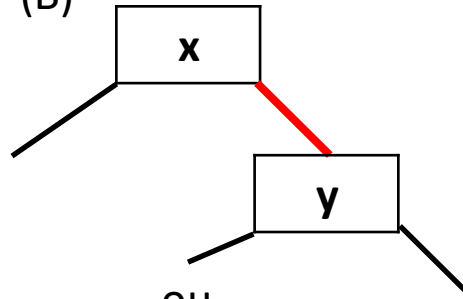
□ **Arbre Rouge et Noir** : Arbre binaire correspondant à un **B-arbre d'ordre 4**.



(A)

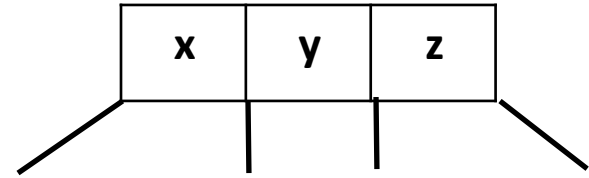
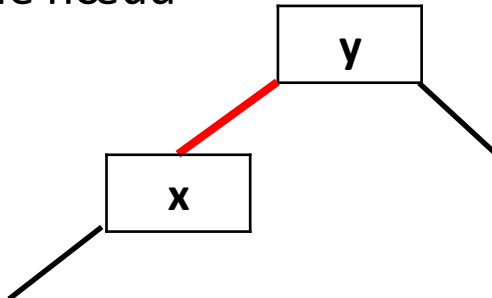


(B)

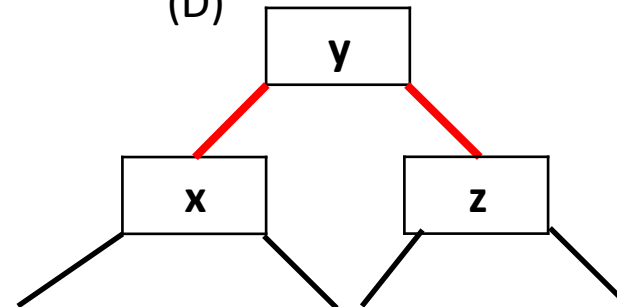


ou

(C)



(D)

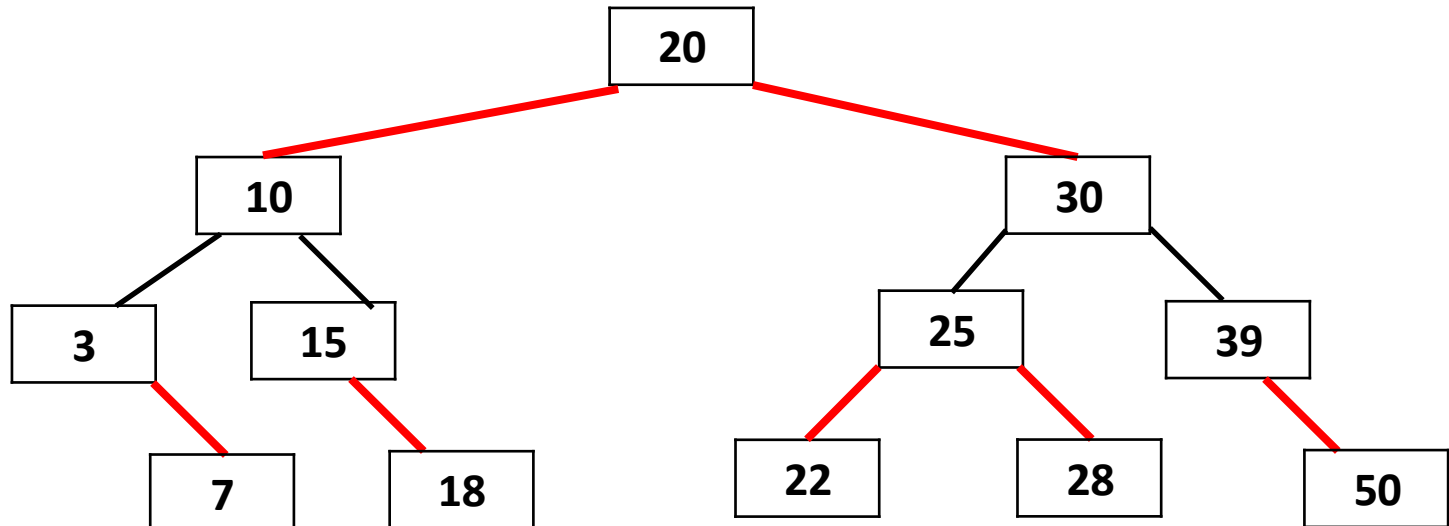
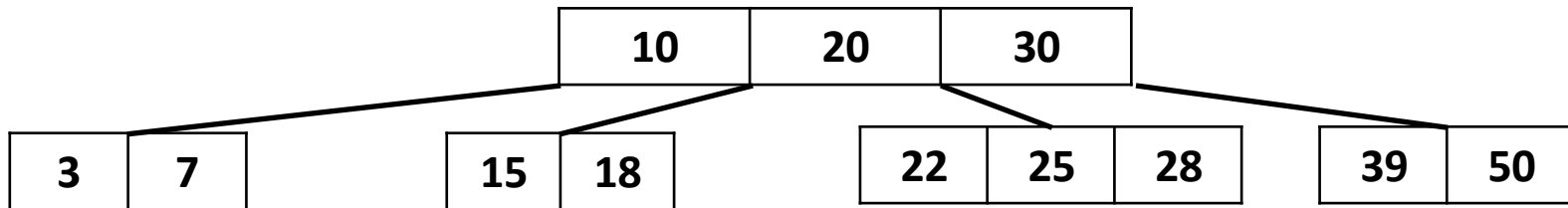


Arcs rouges : lien entre
éléments du même nœud

- Tous les chemins racine → feuille ont le même nombre d'arcs noirs
- On n'a jamais 2 arcs rouges de suite

Arbres Rouge et Noir (bicolore)

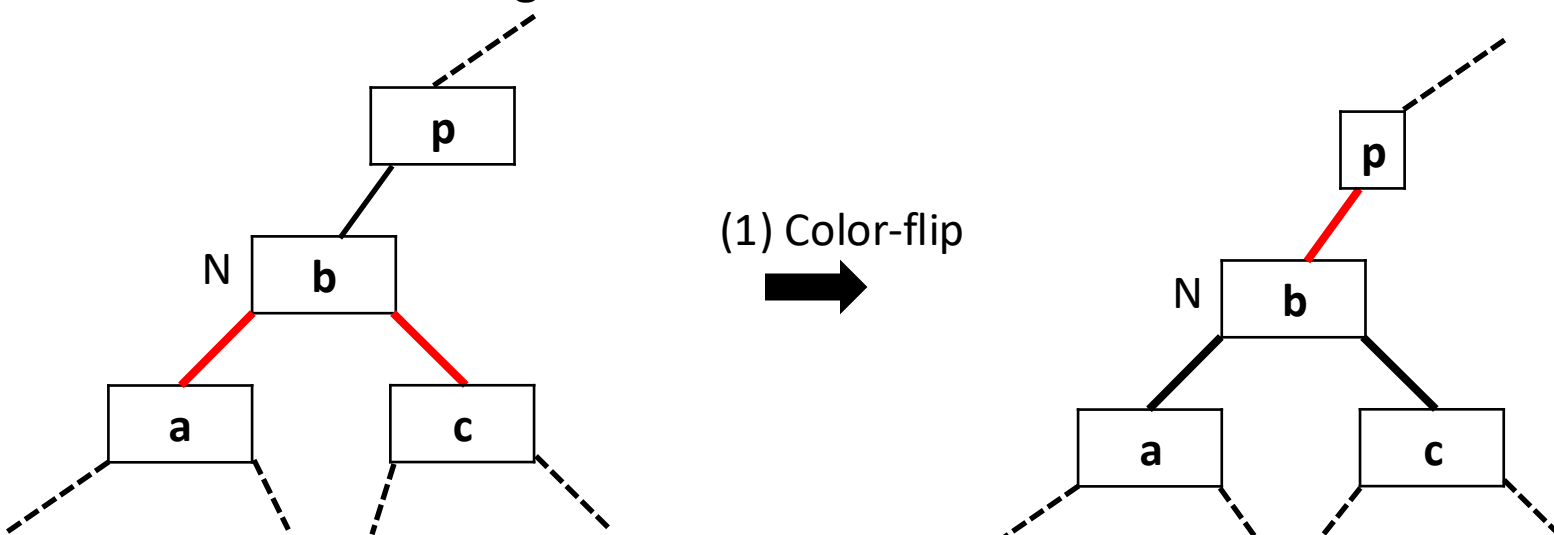
Exemple



Insertion de x dans un ARN

Idée de la taupe: Ne jamais descendre dans un 3-éléments; On divise les nœuds pendant le parcours avant même de faire l'insertion.

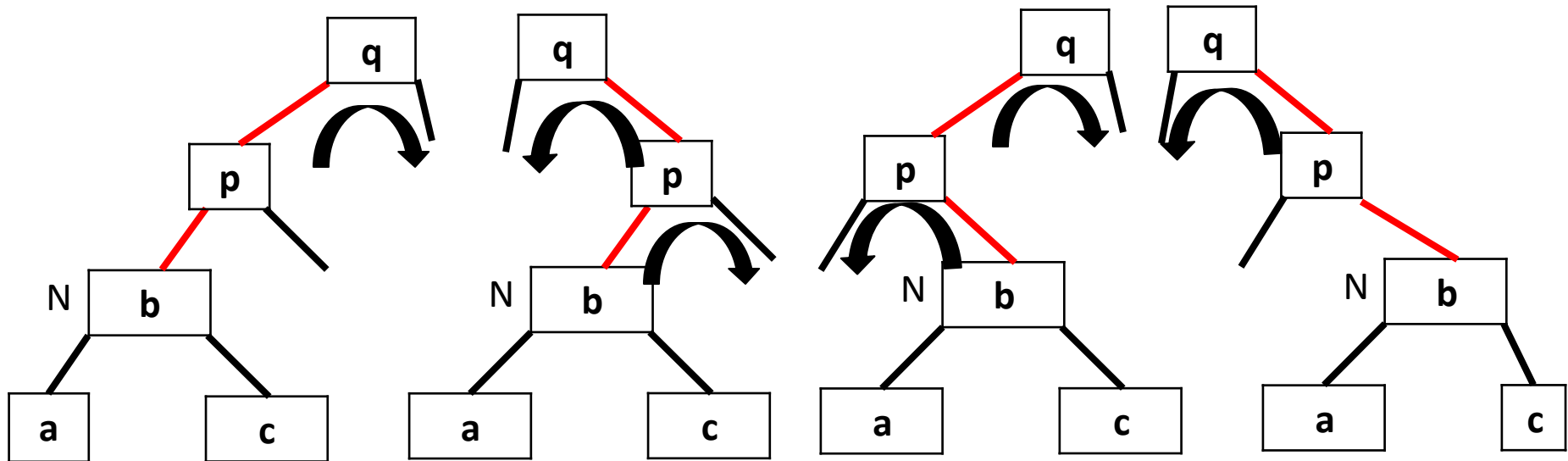
- ❑ À partir de la racine, descendre vers la feuille F où insérer x
- ❑ Pour chaque nœud N rencontré:
 - ❑ Si N est la racine d'un ensemble de 3 éléments:
 - ❑ (1) Faire un color-flip sur N pour le faire monter avec son parent
 - ❑ (2) Si deux arcs rouges se suivent faire **une ou deux rotations** pour arriver à la configuration (D)
(rotation pour aligner les deux arcs + rotation pour arriver à (D))
- ❑ Insérer x avec un arc rouge



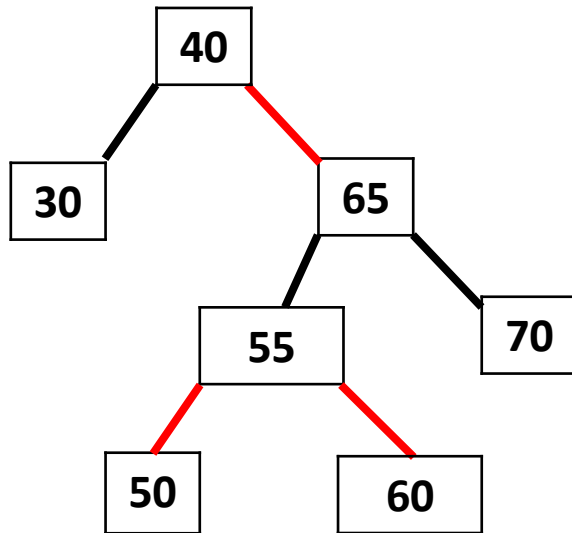
Insertion de x dans un ARN

Idée de la taupe: Ne jamais descendre dans un 3-éléments; On divise les nœuds pendant le parcours avant même de faire l'insertion.

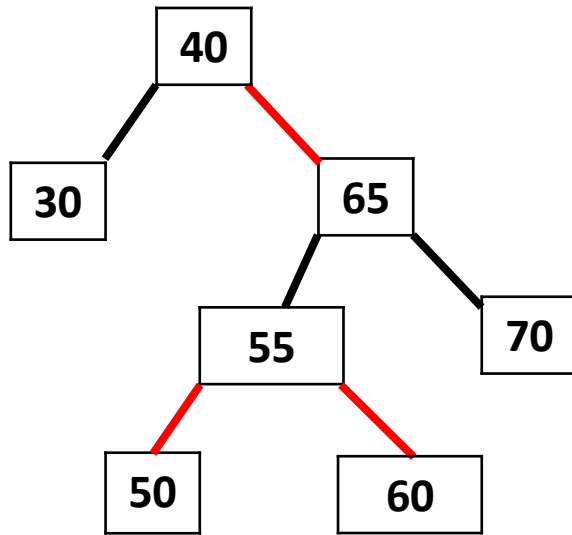
(2) Rotation pour arriver à la configuration (D)



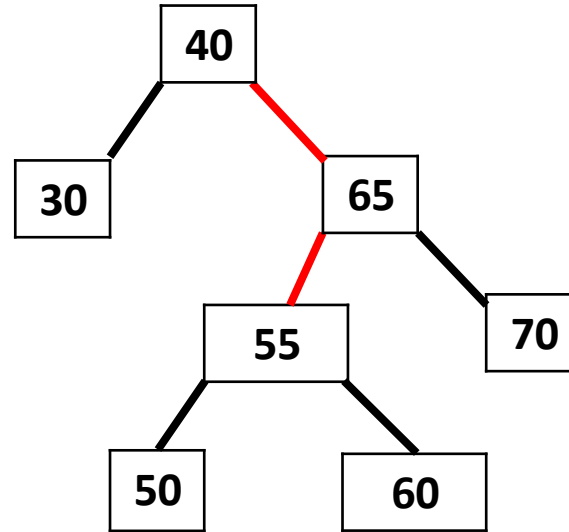
Exemple: Insertion de 52 dans un ARN



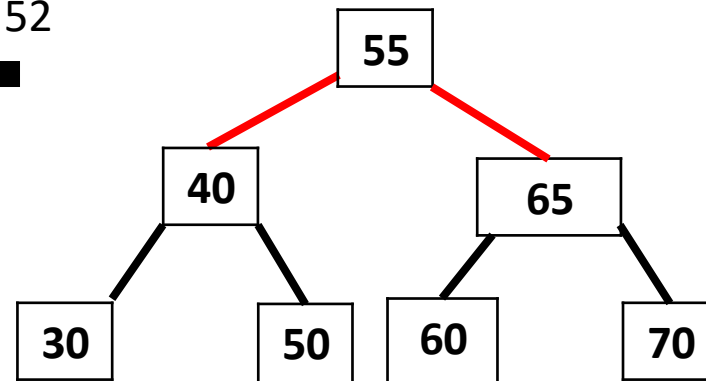
Exemple: Insertion de 52 dans un ARN



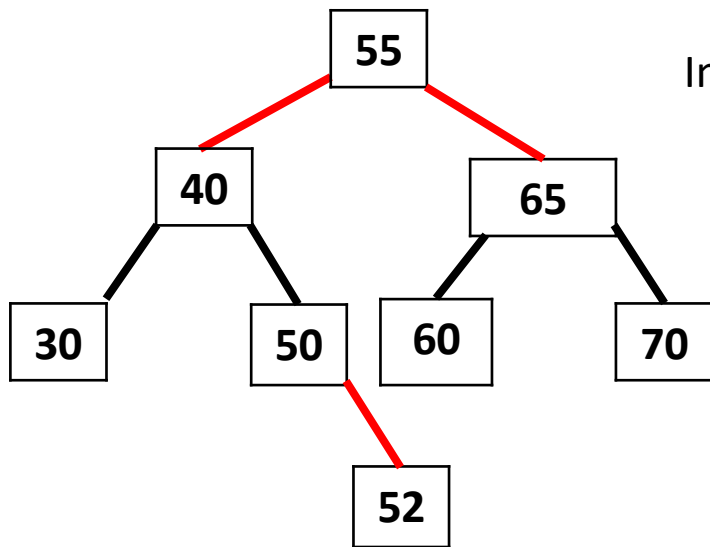
(1) color-flip



(2) 2 rotations pour arriver à la configuration (D)



Insérer 52



Suppression de x dans un ARN

- Même principe que pour tous les arbres binaires de recherche: on se ramène toujours à supprimer une feuille (suppression logique en remplaçant par un autre élément).
- Dans les arbres rouges et noirs on doit seulement supprimer des feuilles dont l'arc est rouge.

- ☐ À partir de la racine, descendre vers la feuille F à supprimer
- ☐ Durant la descente, si on doit descendre dans une arête noire
 - ☐ Si l'arête parent est rouge, faire un color-flip, puis descendre
 - ☐ Sinon, si l'arête sœur est rouge, faire une rotation pour emprunter le rouge, puis descendre
 - ☐ Si aucun rouge nulle part (parent ou soeur), descendre
- ☐ Supprimer F
- ☐ Si F était rattachée par une arête noire
 - ☐ Faire un transfert, ou une fusion