

# Les arbres RB

D.E ZEGOUR

Ecole Supérieure d'Informatique

ESI

# Les arbres RB

## **Introduction**

Structure de données très populaire

Implémentés et intégrés dans plusieurs langages de programmation (JAVA, C...)

Utilisés pour implémenter les dictionnaires et les tableaux associatifs

# Les arbres RB

## Définition

Un arbre rouge et noir (RB-tree) est un arbre binaire de recherche où chaque nœud est de couleur rouge ou noire .

De plus, toutes les branches issues de tout nœud :

1. Ne possèdent pas deux nœuds rouges consécutifs.
2. Possèdent le même nombre de nœuds noirs

Nil considéré comme Noir

- Nœuds noirs : équilibrage parfait
- Nœuds rouges : tolérer légèrement le déséquilibre

Pire des cas:

Alternance entre les nœuds rouges et noirs.

# Les arbres RB

## Insertion

Insertion comme dans un arbre de recherche binaire.

Le nœud inséré est toujours une feuille

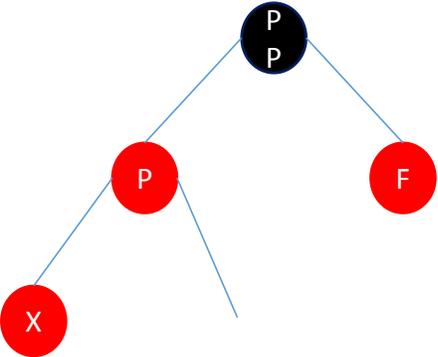
On lui attribue la couleur rouge

Si son père est aussi rouge, un algorithme de maintenance est appliqué

# Les arbres RB

## Insertion / Opérations de maintenance

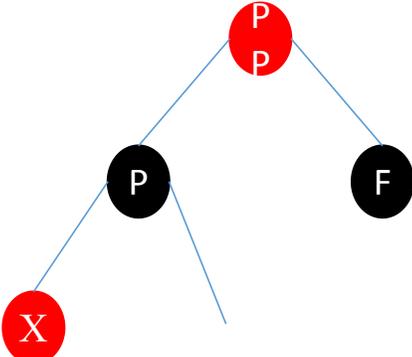
CAS 1: le frère F de P est rouge



X : nœud introduit  
(ou propagé)



Les nœuds P et F deviennent noirs et leur père PP devient rouge.

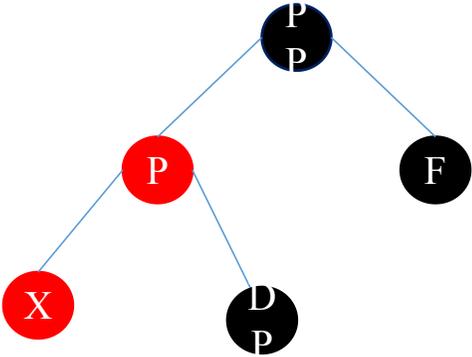


Le processus continue en cascade

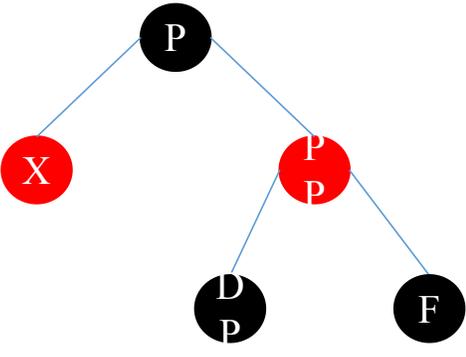
# Les arbres RB

## Insertion / Opérations de maintenance

CAS 2: le frère F de P est noir et X est le fils gauche de P.



Rotation droite du nœud PP.  
P devient noir et PP rouge.

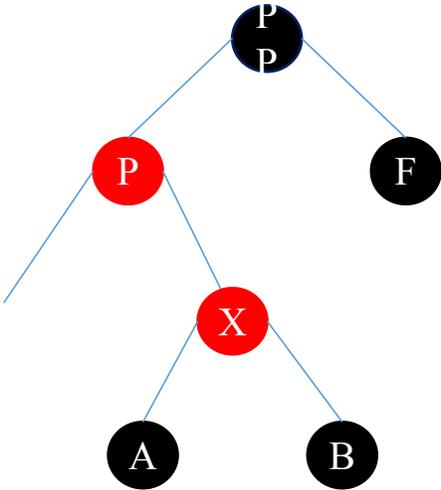


Le processus se termine

# Les arbres RB

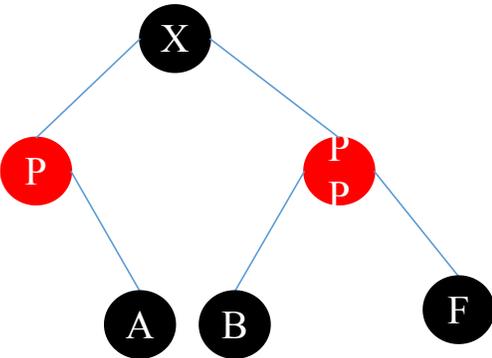
## Insertion / Opérations de maintenance

CAS 3: le frère F de P est noir et X est le fils droit de P.



Rotation gauche du nœud P + rotation droite du nœud PP.

X devient noir et PP rouge.

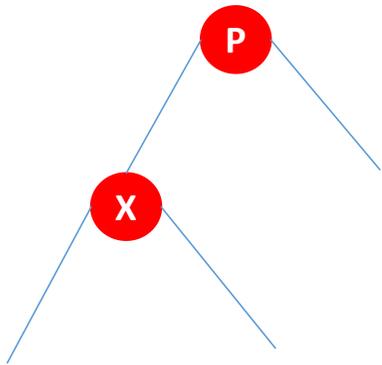


Le processus se termine

# Les arbres RB

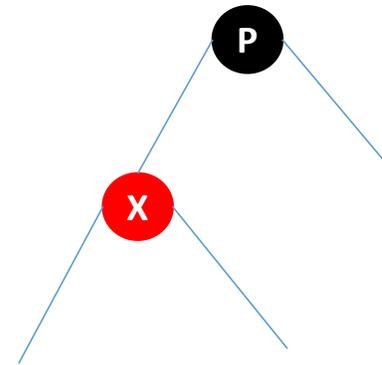
**Insertion** / Opérations de maintenance

CAS 4: le nœud père P est la racine de l'arbre



**Le nœud père devient noir**

**C'est le seul cas où la hauteur  
noire de l'arbre augmente.**



Le processus se termine

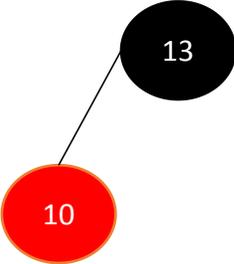
# Les arbres RB

## Insertion / Exemple

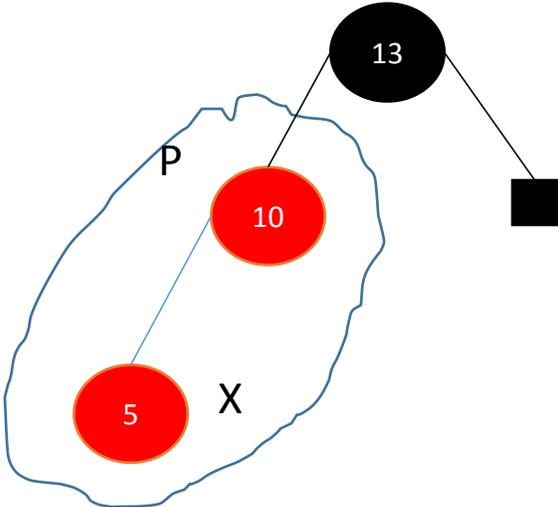
Insérer 13



Insérer 10



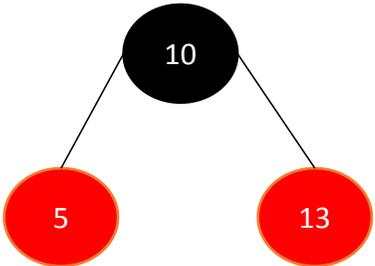
Insérer 5



Couleur du frère de P= noir  
Et X = fg(P)



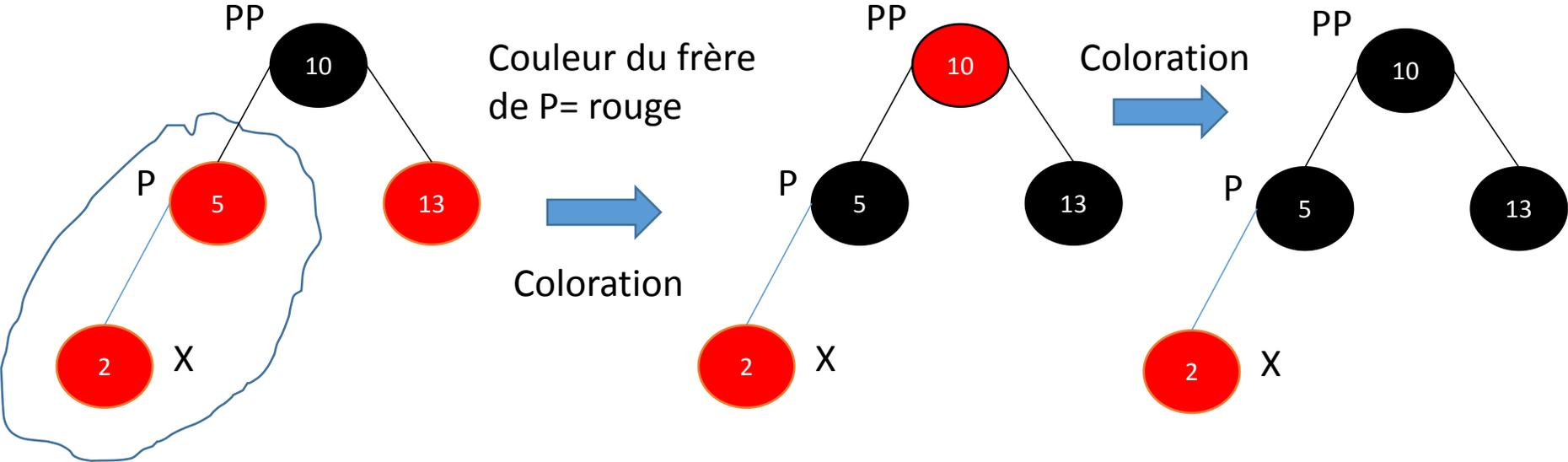
Rotation



# Les arbres RB

## Insertion / Exemple

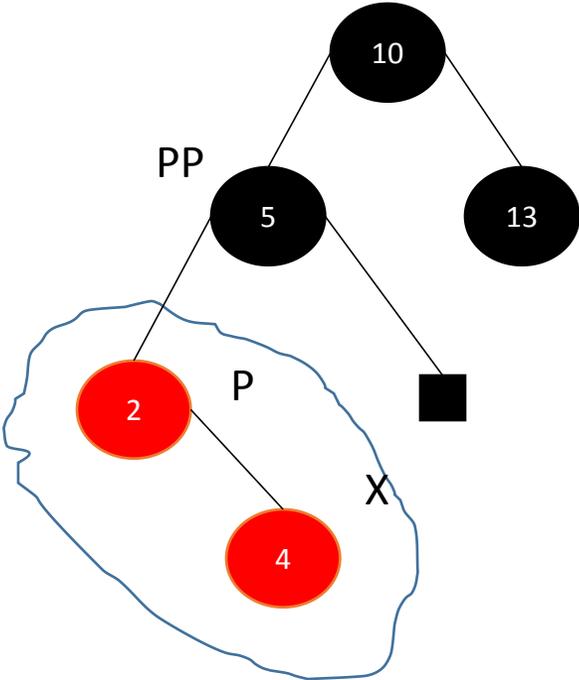
Insérer 2



# Les arbres RB

## Insertion / Exemple

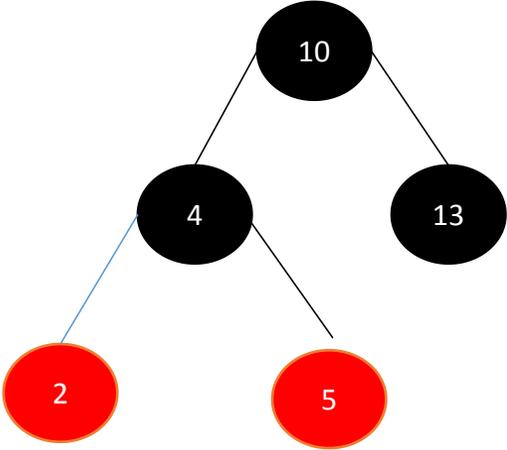
Insérer 4



Couleur du frère de P= noir  
ET X = fd(P) ET P=FG(PP)



Double rotation



# Les arbres RB

## Suppression

Suppression comme dans un arbre de recherche binaire.

Si le nœud physiquement supprimé est noir, un algorithme de maintenance est appliqué.

On considère que le nœud qui remplace le nœud supprimé porte une couleur noire en plus.

Ceci signifie qu'il devient noir s'il est rouge et qu'il devient doublement noir s'il est déjà noir.

L'algorithme de maintenance : éliminer le noeud doublement noir.

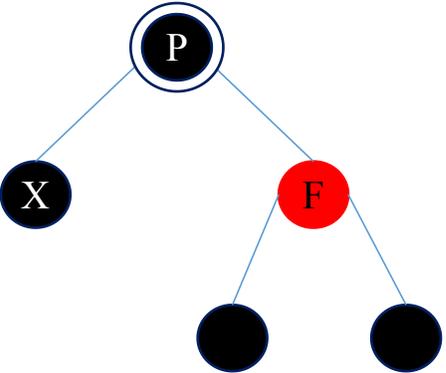
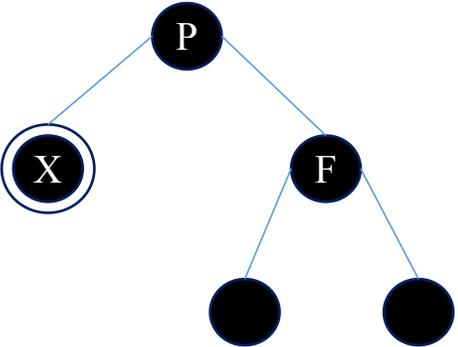
# Les arbres RB

## Suppression / Opération de maintenance

CAS 1: Le frere F de X est noir et a deux fils noirs

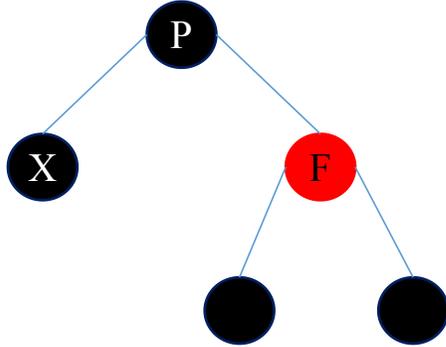
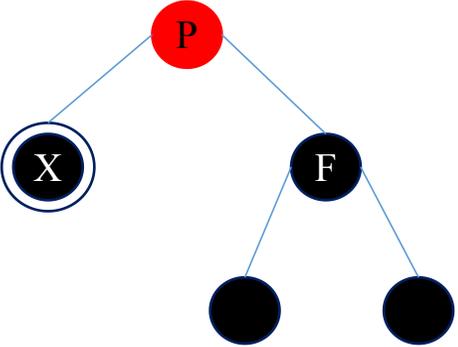
Soit X le noeud doublement noir (Supposé ici comme un fils gauche) et P son père

Cas 1 A



Transformer F en un noeud rouge

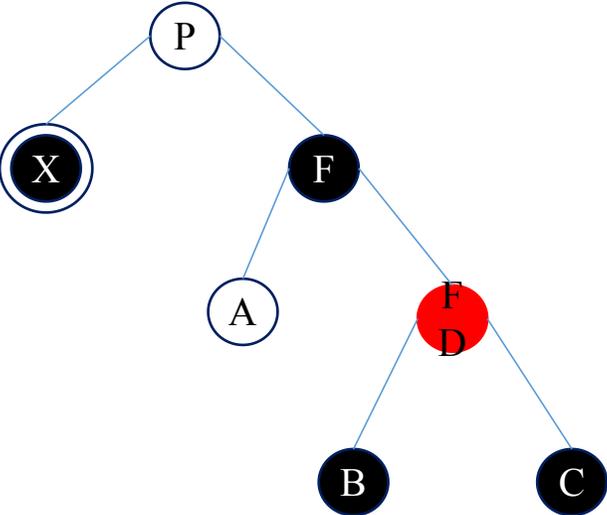
Cas 1 B



Si le parent P est noir, le processus continue en remontant dans l'arbre. P devient le nouveau noeud doublement noir.

# Les arbres RB

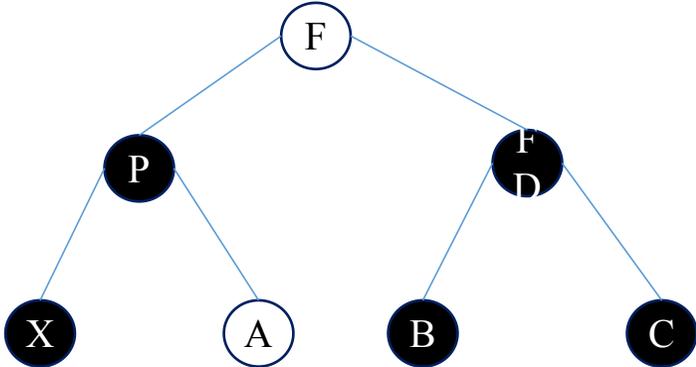
Suppression / Opération de maintenance



Rotation gauche de P

- Recolorer : P et FD deviennent noirs
- La couleur de F est celle de P avant la transformation.

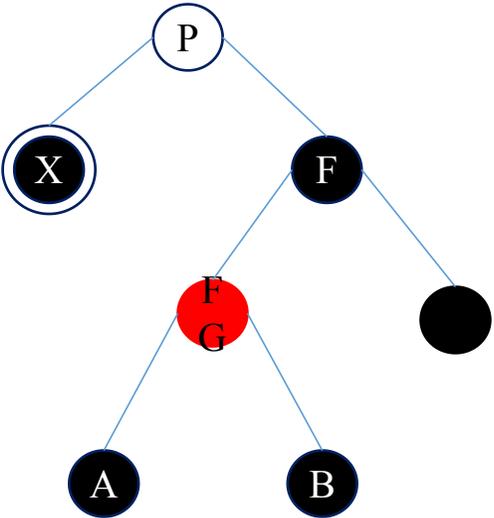
CAS 2: Le frère F du noeud X est noir et a un fils droit rouge (FD)



Le processus se termine

# Les arbres RB

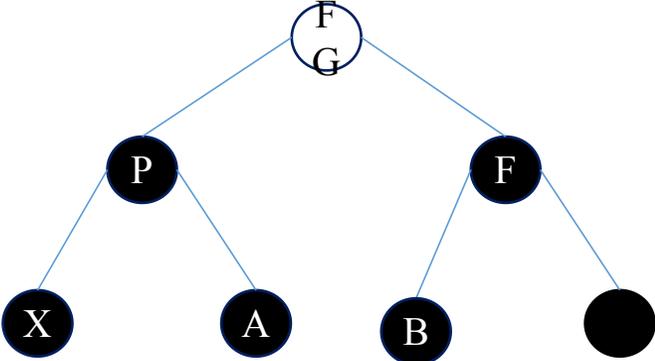
Suppression / Opération de maintenance



Rotation droite (F) +  
Rotation gauche (P).

Recolorer : P devient noir  
La couleur de FG est celle de P  
avant la transformation.

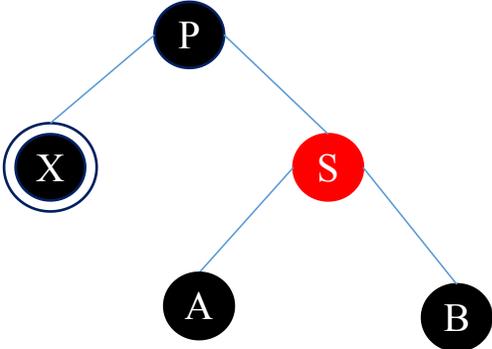
CAS 3: Le frere F du noeud X est noir et a un fils gauche rouge(FG)



Le processus se termine

# Les arbres RB

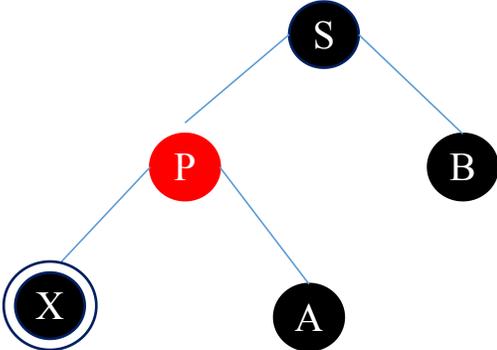
Suppression / Opération de maintenance



Rotation gauche (P)

P devient Rouge et S noir

CAS 4: Le frère F de X est rouge

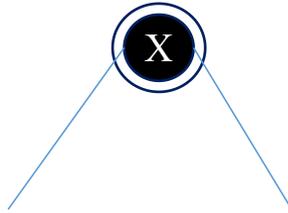


Le processus continue selon le cas 1, 2 ou 3

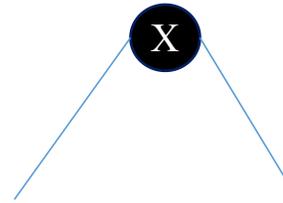
# Les arbres RB

## Suppression / Opération de maintenance

Soit X le noeud doublement noir



X devient simplement un noeud noir.

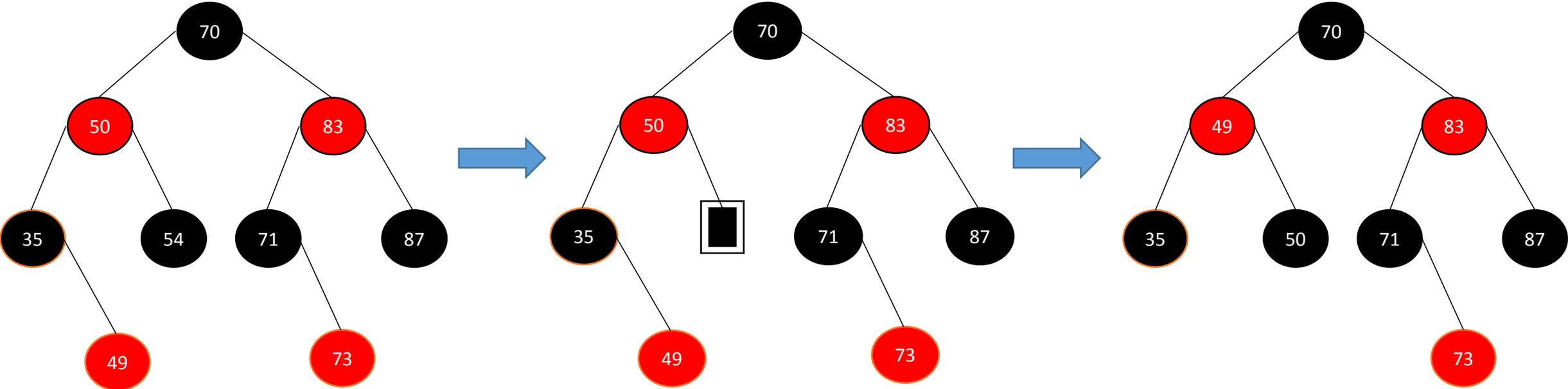


CAS 0: X est la racine de l'arbre

C'est le seul cas où la hauteur de l'arbre diminue. Le processus se termine.

# Les arbres RB

## Suppression / Exemple



Sup 54

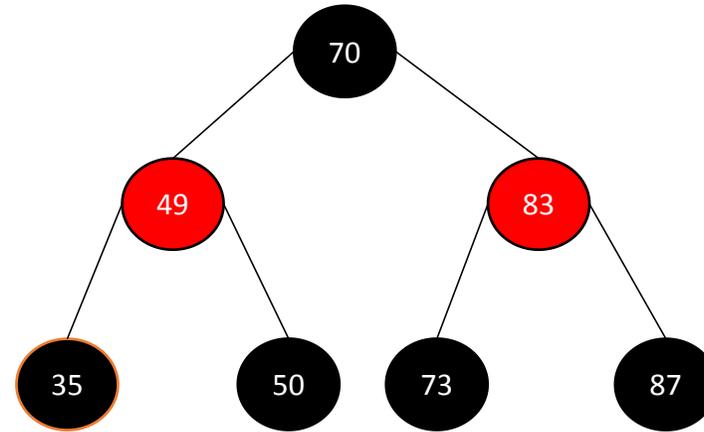
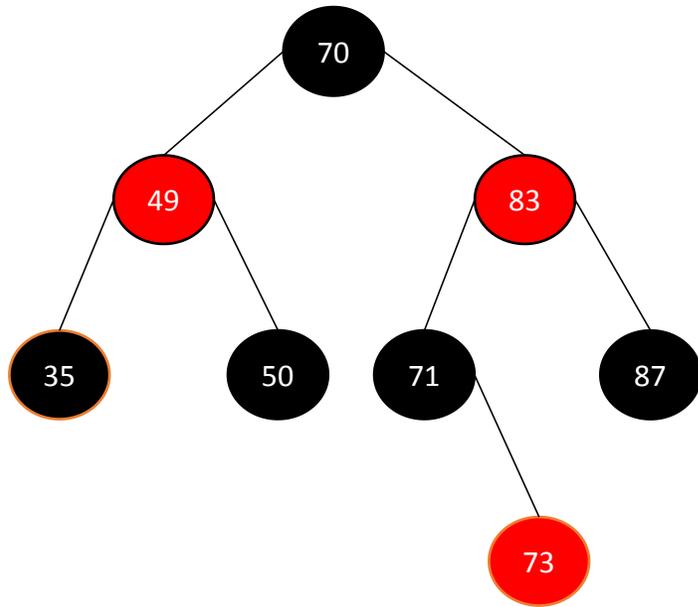
Le nœud qui remplace le nœud supprimé est nil (noir). Lui attribuer une couleur noire en plus.

Cas 3

Rotg(35) + Rotd(50)  
50 devient Noir ; 49 devient Rouge

# Les arbres RB

## Suppression / Exemple

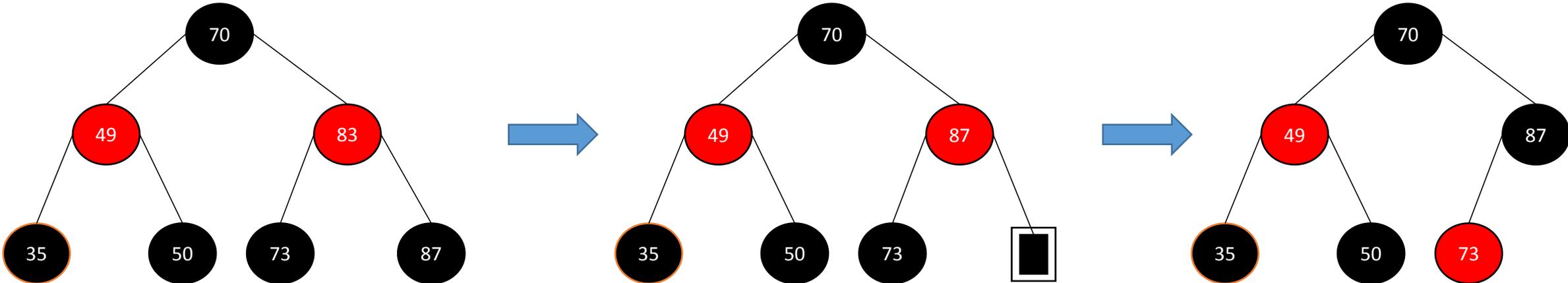


Sup 71

Le nœud qui remplace le nœud supprimé est 73 (rouge). Lui attribuer une couleur noire en plus.

# Les arbres RB

## Suppression / Exemple



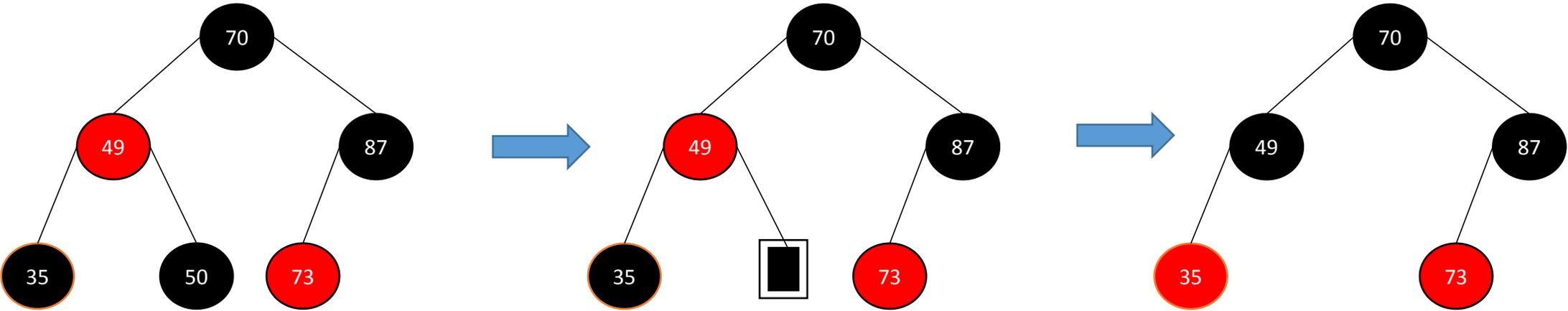
Sup 83

Le nœud qui remplace le nœud supprimé est nil (Noir). Lui attribuer une couleur noire en plus.

Cas 1 B  
73 devient Rouge  
87 devient Noir

# Les arbres RB

## Suppression / Exemple



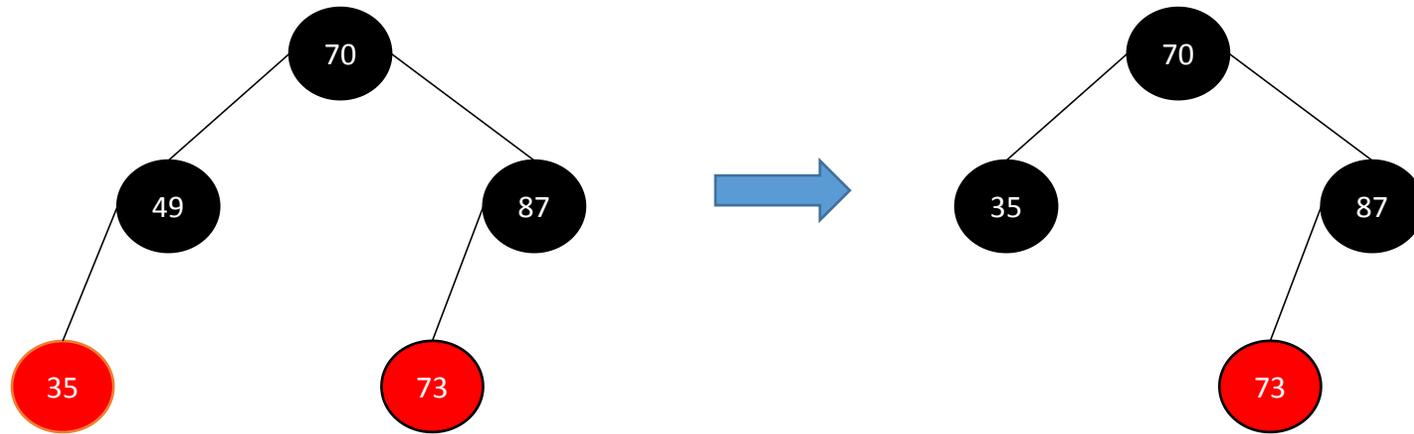
Sup 50

Le nœud qui remplace le nœud physiquement supprimé est nil (Noir)  
Lui attribuer une couleur noire en plus.

Cas 1 B  
35 devient Rouge  
49 devient Noir

# Les arbres RB

## Suppression / Exemple

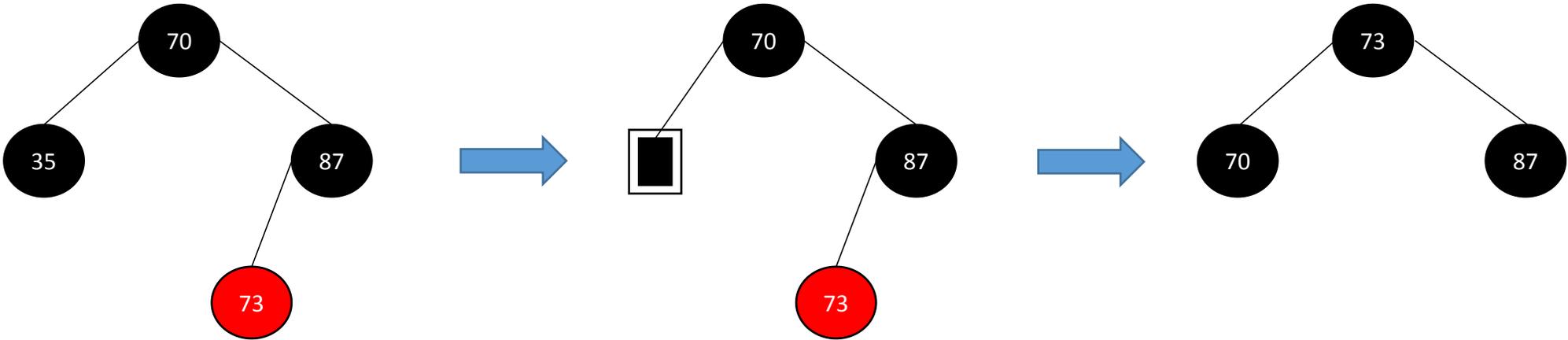


Sup 49

Le nœud qui remplace le nœud supprimé est 35 (Rouge). Lui attribuer une couleur noire en plus.

# Les arbres RB

## Suppression / Exemple



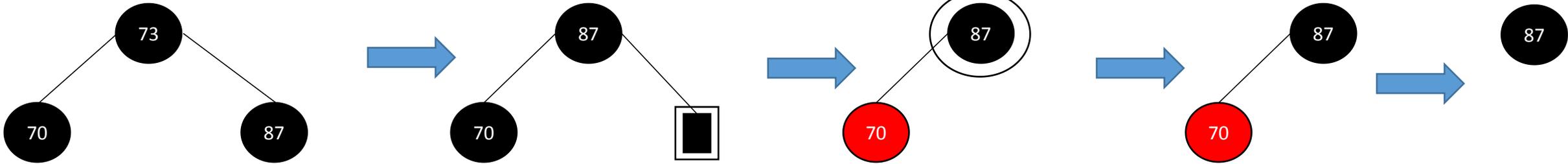
Sup 35

Le nœud qui remplace le nœud physiquement supprimé est nil (Noir). Lui attribuer une couleur noire en plus.

**Cas 3**  
Rotd(87) + Rotg(70)  
70 et 73 deviennent Noir

# Les arbres RB

## Suppression / Exemple



Sup 73

Le nœud qui remplace le nœud physiquement supprimé est nil (Noir). Lui attribuer une couleur noire en plus.

Cas 1 A  
70 devient rouge  
87 devient doublement noir

Sup 70

Le nœud qui remplace le nœud supprimé est 70 ( Rouge).Lui attribuer une couleur noire en plus.

# Les arbres RB

## Arbres RB (Analyse théorique)

la profondeur maximale d'un arbre binaire équilibré est  $2 \cdot \log_2 n$

**La recherche dans un tel arbre** n'exige jamais plus de 100% de plus de comparaisons que pour un arbre binaire complet

### **Operations de maintenance :**

- Restructuration et coloration.
- Insertion : au plus 1 restructuration et au plus  $\log_2 (N)$  colorations.
- suppression : au plus 2 restructurations et au plus  $\log_2 (N)$  colorations.