Algorithmique des structures arborescentes
Algorithmique et programmation fonctionnelle

L2 Info, Math-info, CMI ISI, OPTIM 2022–2023

Marc Zeitoun

2 février 2023

Plan

Familles d'arbres équilibrés

Les AVL

Les arbres rouges et noirs

Plan

ABR : Complexité recherche/insertion/suppression

- Complexité de recherche/insertion/suppression dans un ABR :

ABR : Complexité recherche/insertion/suppression

- Complexité de recherche/insertion/suppression dans un ABR :
  $O(h)$ où h est la **hauteur** de l'arbre.

ABR : Complexité recherche/insertion/suppression

- Complexité de recherche/insertion/suppression dans un ABR :

  $O(h)$ où h est la **hauteur** de l'arbre.

Si n est le nombre de nœuds,

- Au pire, $h =$

ABR : Complexité recherche/insertion/suppression

- Complexité de recherche/insertion/suppression dans un ABR :

  $\boldsymbol{O(h)}$ où $\boldsymbol{h}$ est la **hauteur** de l'arbre.

Si $\boldsymbol{n}$ est le nombre de nœuds,

- Au pire, $\boldsymbol{h = n - 1}$.
- Au mieux, $\boldsymbol{h =}$

ABR : Complexité recherche/insertion/suppression

- Complexité de recherche/insertion/suppression dans un ABR :

$$\boldsymbol{O(h)} \text{ où } \boldsymbol{h} \text{ est la } \textbf{hauteur} \text{ de l'arbre.}$$

Si $\boldsymbol{n}$ est le nombre de nœuds,

- Au pire, $\boldsymbol{h = n - 1}$.
- Au mieux, $\boldsymbol{h = \log_2(n+1) - 1}$.

Est-ce que le mauvais cas, $\boldsymbol{h = n - 1}$, peut être obtenu par insertions ?

Familles d'arbres équilibrés

Problème : si on insère 1, puis 2, 3, 4, ..., n, on obtient un arbre filiforme.

Familles d'arbres équilibrés

Problème : si on insère 1, puis 2, 3, 4, ..., n, on obtient un arbre filiforme.

- Maintenir dans les ABR

$$\boldsymbol{h = \alpha \log(n)}$$

garantirait une complexité $\boldsymbol{O(\log(n))}$ **pour ces opérations**.

Familles d'arbres équilibrés

Problème : si on insère 1, puis 2, 3, 4, ..., n, on obtient un arbre filiforme.

- Maintenir dans les ABR

$$\boldsymbol{h = \alpha \log(n)}$$

garantirait une complexité $\boldsymbol{O(\log(n))}$ **pour ces opérations**.

Nouvel objectif

Maintenir $\boldsymbol{h = \alpha \log(n)}$ pour un nombre $\boldsymbol{\alpha > 0}$.

Plan

Familles d'arbres équilibrés

Les AVL

Les arbres rouges et noirs

Maintien de l'équilibre : les rotations

Les AVL (Adelson-Velsky et Landis, 1962)

Arbre parfait : chacun des 2 sous-arbres de chaque nœud a 50% des nœuds

Les AVL (Adelson-Velsky et Landis, 1962)

Arbre parfait : chacun des 2 sous-arbres de chaque nœud a 50% des nœuds

Intuition. Même sans couper exactement en 2, si chaque sous-arbre a un pourcentage minimal des nœuds, la hauteur sera logarithmique.

Les AVL (Adelson-Velsky et Landis, 1962)

Arbre parfait : chacun des 2 sous-arbres de chaque nœud a 50% des nœuds

Intuition. Même sans couper exactement en 2, si chaque sous-arbre a un pourcentage minimal des nœuds, la hauteur sera logarithmique.

Trouver une condition garantissant $> x\%$, au lieu de $= 50\%$.

Les AVL (Adelson-Velsky et Landis, 1962)

Arbre parfait : chacun des 2 sous-arbres de chaque nœud a 50% des nœuds

Intuition. Même sans couper exactement en 2, si chaque sous-arbre a un pourcentage minimal des nœuds, la hauteur sera logarithmique.

Trouver une condition garantissant $> x\%$, au lieu de $= 50\%$.

- **Équilibre** d'un nœud $=$

  hauteur(sous-arbre gauche) $-$ hauteur(sous-arbre droit).
- **AVL** $=$ ABR dont chaque nœud a un équilibre $-1, 0$ ou 1.

Relation taille-hauteur pour les AVL

Soit $N(h)$ le nombre minimal de nœuds d'un AVL de hauteur h.

Plus petits AVL de hauteur donnée :

- Hauteur 0 :

Relation taille-hauteur pour les AVL

Soit $N(h)$ le nombre minimal de nœuds d'un AVL de hauteur h.

Plus petits AVL de hauteur donnée :

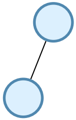
- Hauteur 0 : 1 nœud. Donc $N(0) = 1$.
- Hauteur 1 :

Relation taille-hauteur pour les AVL

Soit $N(h)$ le nombre minimal de nœuds d'un AVL de hauteur h.

Plus petits AVL de hauteur donnée :

- Hauteur 0 : 1 nœud. Donc $N(0) = 1$.
- Hauteur 1 : 2 nœuds. Donc $N(1) = 2$.



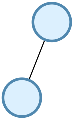
- hauteur 2 :

Relation taille-hauteur pour les AVL

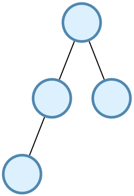
Soit $N(h)$ le nombre minimal de nœuds d'un AVL de hauteur h.

Plus petits AVL de hauteur donnée :

- Hauteur 0 : 1 nœud. Donc $N(0) = 1$.
- Hauteur 1 : 2 nœuds. Donc $N(1) = 2$.



- hauteur 2 : 4 nœuds. Donc $N(2) = 4$.



Relation taille-hauteur pour les AVL

Soit $N(h)$ le nombre minimal de nœuds d'un AVL de hauteur h.

Si t est un AVL de hauteur h et de taille minimale :

1. l'un des sous-arbres est un AVL **de hauteur**

Relation taille-hauteur pour les AVL

Soit $N(h)$ le nombre minimal de nœuds d'un AVL de hauteur h.

Si t est un AVL de hauteur h et de taille minimale :

1. l'un des sous-arbres est un AVL **de hauteur** $h-1$ de taille minimale,

Relation taille-hauteur pour les AVL

Soit $N(h)$ le nombre minimal de nœuds d'un AVL de hauteur h.

Si t est un AVL de hauteur h et de taille minimale :

1. l'un des sous-arbres est un AVL **de hauteur** $h-1$ de taille minimale,
2. l'autre est un AVL **de hauteur**

Relation taille-hauteur pour les AVL

Soit $N(h)$ le nombre minimal de nœuds d'un AVL de hauteur h.

Si t est un AVL de hauteur h et de taille minimale :

1. l'un des sous-arbres est un AVL **de hauteur** $h-1$ de taille minimale,
2. l'autre est un AVL **de hauteur** $h-2$ de taille minimale.

Relation taille-hauteur pour les AVL

Soit $N(h)$ le nombre minimal de nœuds d'un AVL de hauteur h.

Si t est un AVL de hauteur h et de taille minimale :

1. l'un des sous-arbres est un AVL **de hauteur** $h-1$ de taille minimale,
2. l'autre est un AVL **de hauteur** $h-2$ de taille minimale.

Donc $N(h) = 1 + N(h-1) + N(h-2)$.

Relation taille-hauteur pour les AVL

Soit $N(h)$ le nombre minimal de nœuds d'un AVL de hauteur h.

Si t est un AVL de hauteur h et de taille minimale :

1. l'un des sous-arbres est un AVL **de hauteur** $h-1$ de taille minimale,
2. l'autre est un AVL **de hauteur** $h-2$ de taille minimale.

Donc $N(h) = 1 + N(h-1) + N(h-2)$.

Soit $\phi = (1+\sqrt{5})/2 \approx 1.61$. On a $\phi^2 = \phi + 1$.

On montre par récurrence que $\boxed{N(h) \geqslant \phi^h - 1}$.

Relation taille-hauteur pour les AVL

On a $N(0) = 1$, $N(1) = 2$ et $N(h) = 1 + N(h-1) + N(h-2)$.

On montre par récurrence que $\boxed{N(h) \geqslant \phi^h - 1}$.

Relation taille-hauteur pour les AVL

On a $N(0) = 1$, $N(1) = 2$ et $N(h) = 1 + N(h-1) + N(h-2)$.

On montre par récurrence que $\boxed{N(h) \geqslant \phi^h - 1}$.

- $h = 0$: $N(0) = 1$ et $\phi^0 - 1 = 1 - 1 = 0$, **OK**.

Relation taille-hauteur pour les AVL

On a $N(0) = 1$, $N(1) = 2$ et $N(h) = 1 + N(h-1) + N(h-2)$.

On montre par récurrence que $\boxed{N(h) \geqslant \phi^h - 1}$.

- $h = 0$: $N(0) = 1$ et $\phi^0 - 1 = 1 - 1 = 0$, **OK**.
- $h = 1$: $N(1) = 2$ et $\phi^1 - 1 \simeq 0.61$, **OK**.

Relation taille-hauteur pour les AVL

On a $N(0) = 1$, $N(1) = 2$ et $N(h) = 1 + N(h-1) + N(h-2)$.

On montre par récurrence que $\boxed{N(h) \geqslant \phi^h - 1}$.

- $h = 0$: $N(0) = 1$ et $\phi^0 - 1 = 1 - 1 = 0$, **OK**.
- $h = 1$: $N(1) = 2$ et $\phi^1 - 1 \simeq 0.61$, **OK**.
- **Supposons la propriété vraie jusqu'au rang $h-1$.**

  Montrons-la pour h :

Relation taille-hauteur pour les AVL

On a $N(0) = 1$, $N(1) = 2$ et $N(h) = 1 + N(h-1) + N(h-2)$.

On montre par récurrence que $\boxed{N(h) \geqslant \phi^h - 1}$.

- $h = 0$: $N(0) = 1$ et $\phi^0 - 1 = 1 - 1 = 0$, **OK**.
- $h = 1$: $N(1) = 2$ et $\phi^1 - 1 \simeq 0.61$, **OK**.
- **Supposons la propriété vraie jusqu'au rang $h-1$.**
  Montrons-la pour h :

$$N(h) = 1 + N(h-1) + N(h-2)$$

Relation taille-hauteur pour les AVL

On a $N(0) = 1$, $N(1) = 2$ et $N(h) = 1 + N(h-1) + N(h-2)$.

On montre par récurrence que $\boxed{N(h) \geqslant \phi^h - 1}$.

- $h = 0$: $N(0) = 1$ et $\phi^0 - 1 = 1 - 1 = 0$, **OK**.
- $h = 1$: $N(1) = 2$ et $\phi^1 - 1 \simeq 0.61$, **OK**.
- **Supposons la propriété vraie jusqu'au rang $h-1$.**
  Montrons-la pour h :

$$\begin{aligned} N(h) &= 1 + N(h-1) + N(h-2) \\ &\geqslant 1 + (\phi^{h-1} - 1) + (\phi^{h-2} - 1) \quad \text{par hypothèse de récurrence} \end{aligned}$$

Relation taille-hauteur pour les AVL

On a $N(0) = 1$, $N(1) = 2$ et $N(h) = 1 + N(h-1) + N(h-2)$.

On montre par récurrence que $\boxed{N(h) \geqslant \phi^h - 1}$.

- $h = 0$: $N(0) = 1$ et $\phi^0 - 1 = 1 - 1 = 0$, **OK**.
- $h = 1$: $N(1) = 2$ et $\phi^1 - 1 \simeq 0.61$, **OK**.
- **Supposons la propriété vraie jusqu'au rang $h-1$.**

  Montrons-la pour h :

$$
\begin{aligned}
N(h) &= 1 + N(h-1) + N(h-2) \\
&\geqslant 1 + (\phi^{h-1} - 1) + (\phi^{h-2} - 1) \quad \text{par hypothèse de récurrence} \\
&\geqslant (\phi^{h-1} + \phi^{h-2}) - 1
\end{aligned}
$$

Relation taille-hauteur pour les AVL

On a $N(0) = 1$, $N(1) = 2$ et $N(h) = 1 + N(h-1) + N(h-2)$.

On montre par récurrence que $\boxed{N(h) \geqslant \phi^h - 1}$.

- $h = 0$: $N(0) = 1$ et $\phi^0 - 1 = 1 - 1 = 0$, **OK**.
- $h = 1$: $N(1) = 2$ et $\phi^1 - 1 \simeq 0.61$, **OK**.
- **Supposons la propriété vraie jusqu'au rang $h-1$.**

  Montrons-la pour h :

$$\begin{aligned} N(h) &= 1 + N(h-1) + N(h-2) \\ &\geqslant 1 + (\phi^{h-1} - 1) + (\phi^{h-2} - 1) \quad \text{par hypothèse de récurrence} \\ &\geqslant (\phi^{h-1} + \phi^{h-2}) - 1 \\ &\geqslant (\phi^{h-2}(\phi + 1)) - 1 \end{aligned}$$

Relation taille-hauteur pour les AVL

On a $N(0) = 1$, $N(1) = 2$ et $N(h) = 1 + N(h-1) + N(h-2)$.

On montre par récurrence que $\boxed{N(h) \geqslant \phi^h - 1}$.

- $h = 0$: $N(0) = 1$ et $\phi^0 - 1 = 1 - 1 = 0$, **OK**.
- $h = 1$: $N(1) = 2$ et $\phi^1 - 1 \simeq 0.61$, **OK**.
- **Supposons la propriété vraie jusqu'au rang $h-1$.**

  Montrons-la pour h :

$$
\begin{aligned}
N(h) &= 1 + N(h-1) + N(h-2) \\
&\geqslant 1 + (\phi^{h-1} - 1) + (\phi^{h-2} - 1) && \text{par hypothèse de récurrence} \\
&\geqslant (\phi^{h-1} + \phi^{h-2}) - 1 \\
&\geqslant (\phi^{h-2}(\phi + 1)) - 1 \\
&\geqslant (\phi^{h-2}(\phi^2)) - 1 && \text{par définition de } \phi
\end{aligned}
$$

Relation taille-hauteur pour les AVL

On a $N(0) = 1$, $N(1) = 2$ et $N(h) = 1 + N(h-1) + N(h-2)$.

On montre par récurrence que $\boxed{N(h) \geqslant \phi^h - 1}$.

- $h = 0$: $N(0) = 1$ et $\phi^0 - 1 = 1 - 1 = 0$, **OK**.
- $h = 1$: $N(1) = 2$ et $\phi^1 - 1 \simeq 0.61$, **OK**.
- **Supposons la propriété vraie jusqu'au rang $h-1$.**
  Montrons-la pour h :

$$\begin{aligned} N(h) &= 1 + N(h-1) + N(h-2) \\ &\geqslant 1 + (\phi^{h-1} - 1) + (\phi^{h-2} - 1) && \text{par hypothèse de récurrence} \\ &\geqslant (\phi^{h-1} + \phi^{h-2}) - 1 \\ &\geqslant (\phi^{h-2}(\phi + 1)) - 1 \\ &\geqslant (\phi^{h-2}(\phi^2)) - 1 && \text{par définition de } \phi \\ &\geqslant \phi^h - 1 \end{aligned}$$

Relation taille-hauteur pour les AVL

On a $N(0) = 1$, $N(1) = 2$ et $N(h) = 1 + N(h-1) + N(h-2)$.

On montre par récurrence que $\boxed{N(h) \geqslant \phi^h - 1}$.

- $h = 0$: $N(0) = 1$ et $\phi^0 - 1 = 1 - 1 = 0$, **OK**.
- $h = 1$: $N(1) = 2$ et $\phi^1 - 1 \simeq 0.61$, **OK**.
- **Supposons la propriété vraie jusqu'au rang $h-1$.**

  Montrons-la pour h :

$$
\begin{aligned}
N(h) &= 1 + N(h-1) + N(h-2) \\
&\geqslant 1 + (\phi^{h-1} - 1) + (\phi^{h-2} - 1) && \text{par hypothèse de récurrence} \\
&\geqslant (\phi^{h-1} + \phi^{h-2}) - 1 \\
&\geqslant (\phi^{h-2}(\phi + 1)) - 1 \\
&\geqslant (\phi^{h-2}(\phi^2)) - 1 && \text{par définition de } \phi \\
&\geqslant \phi^h - 1
\end{aligned}
$$

  OK

Relation taille-hauteur pour les AVL

Soit $N(h)$ le nombre minimal de nœuds d'un AVL de hauteur h.

On a montré que $N(h) \geqslant \phi^h - 1$.

Donc si un AVL de hauteur h a n nœuds :

$$n \geqslant \phi^h - 1.$$

Relation taille-hauteur pour les AVL

Soit $N(h)$ le nombre minimal de nœuds d'un AVL de hauteur h.

On a montré que $N(h) \geqslant \phi^h - 1$.

Donc si un AVL de hauteur h a n nœuds :

$$n \geqslant \phi^h - 1.$$

$$\phi^h \leqslant n + 1.$$

Relation taille-hauteur pour les AVL

Soit $N(h)$ le nombre minimal de nœuds d'un AVL de hauteur h.

On a montré que $N(h) \geqslant \phi^h - 1$.

Donc si un AVL de hauteur h a n nœuds :

$$\begin{aligned} n &\geqslant \phi^h - 1. \\ \phi^h &\leqslant n + 1. \\ h &\leqslant \log_\phi(n+1) && \text{par passage au } \log_\phi \end{aligned}$$

Relation taille-hauteur pour les AVL

Soit $N(h)$ le nombre minimal de nœuds d'un AVL de hauteur h.

On a montré que $N(h) \geqslant \phi^h - 1$.

Donc si un AVL de hauteur h a n nœuds :

$$\begin{aligned}
n &\geqslant \phi^h - 1. \\
\phi^h &\leqslant n + 1. \\
h &\leqslant \log_\phi(n+1) && \text{par passage au } \log_\phi \\
h &\leqslant \frac{1}{\log_2 \phi} \log_2(n+1) && \text{car } \log_\phi(x) = \frac{\log_2 x}{\log_2 \phi}
\end{aligned}$$

Relation taille-hauteur pour les AVL

Soit $N(h)$ le nombre minimal de nœuds d'un AVL de hauteur h.

On a montré que $N(h) \geqslant \phi^h - 1$.

Donc si un AVL de hauteur h a n nœuds :

$$\begin{aligned}
n &\geqslant \phi^h - 1. \\
\phi^h &\leqslant n + 1. \\
h &\leqslant \log_\phi(n+1) && \text{par passage au } \log_\phi \\
h &\leqslant \frac{1}{\log_2 \phi} \log_2(n+1) && \text{car } \log_\phi(x) = \frac{\log_2 x}{\log_2 \phi} \\
h &\leqslant 1.45 \log_2(n+1) && \text{car } 1/\log_2(\phi) \simeq 1.4404
\end{aligned}$$

Relation taille-hauteur pour les AVL

Soit $N(h)$ le nombre minimal de nœuds d'un AVL de hauteur h.

On a montré que $N(h) \geqslant \phi^h - 1$.

Donc si un AVL de hauteur h a n nœuds :

$$
\begin{aligned}
n &\geqslant \phi^h - 1. \\
\phi^h &\leqslant n + 1. \\
h &\leqslant \log_\phi(n+1) && \text{par passage au } \log_\phi \\
h &\leqslant \frac{1}{\log_2 \phi} \log_2(n+1) && \text{car } \log_\phi(x) = \frac{\log_2 x}{\log_2 \phi} \\
h &\leqslant 1.45 \log_2(n+1) && \text{car } 1/\log_2(\phi) \simeq 1.4404
\end{aligned}
$$

On a donc bien $h = O(\log n)$.

Insertion dans les AVL

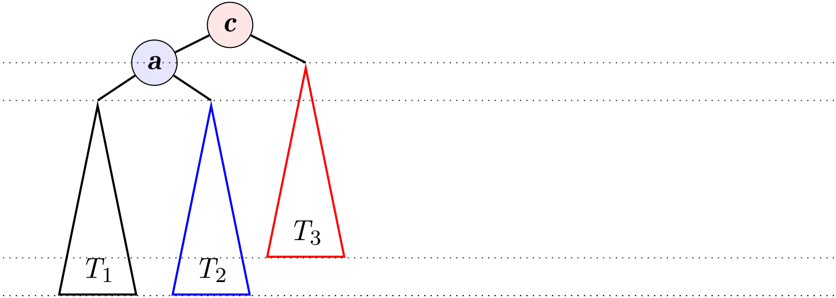
On maintient à la fois la propriété ABR et la propriété AVL.

- Insertion : on commence par insérer comme dans un **ABR**.
- Si création d'un nœud x, l'arbre n'est peut-être plus un **AVL**.
  Un des nœuds a pu passer d'équilibre $\pm\mathbf{1}$ à $\pm\mathbf{2}$.
- $c = \mathbf{1}^{\text{er}}$ nœud sur la branche de x à la racine d'équilibre $\pm\mathbf{2}$.
- On doit **réparer** l'AVL pour rétablir les propriétés.

AVL : équilibrage, cas 1a

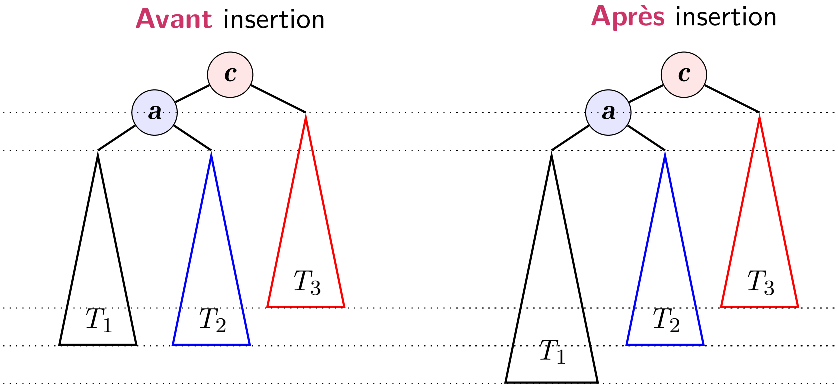
- Insertion dans le sous-arbre gauche du sous-arbre gauche (GG).

Avant insertion



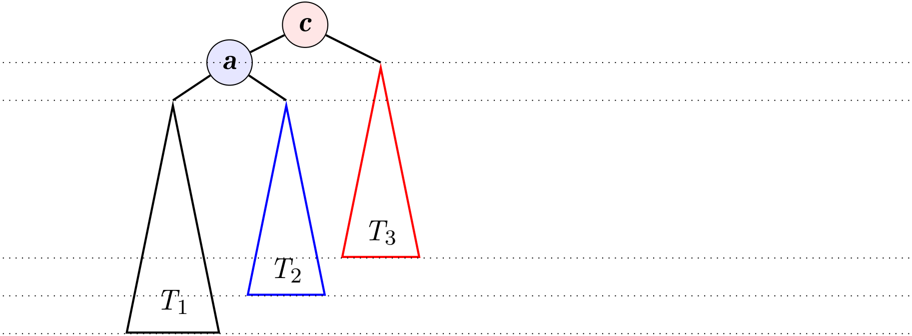
AVL : équilibrage, cas 1a

- Insertion dans le sous-arbre gauche du sous-arbre gauche (GG).



AVL : équilibrage, cas 1a

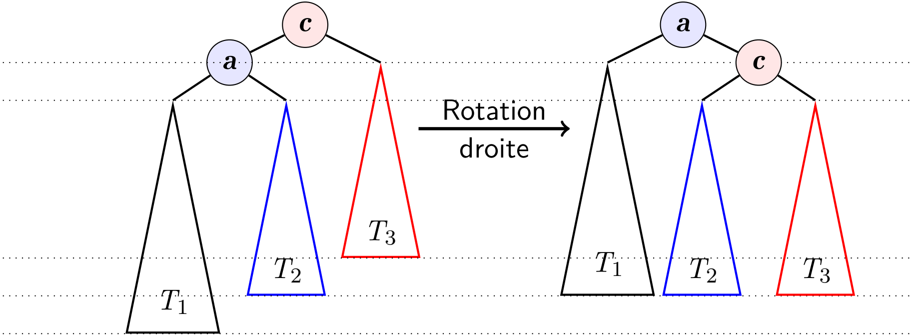
- hauteur(GG) $=$ hauteur(D) $+ 1$



$$\text{clés}(\boldsymbol{T_1}) \leqslant \boldsymbol{a} \leqslant \text{clés}(\boldsymbol{T_2}) \leqslant \boldsymbol{c} \leqslant \text{clés}(\boldsymbol{T_3})$$

AVL : équilibrage, cas 1a

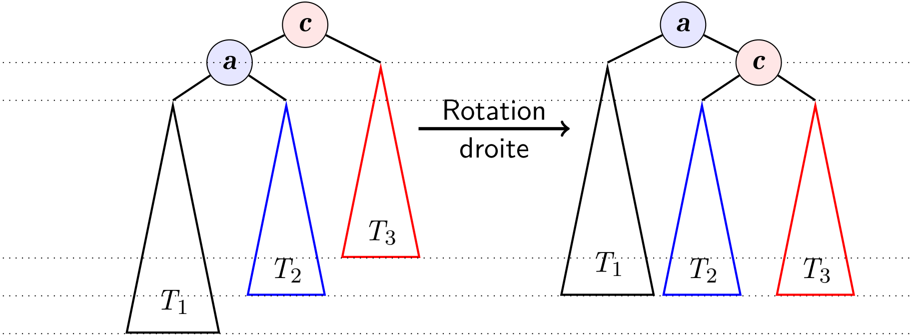
- hauteur(GG) $=$ hauteur(D) $+ 1$



clés$(\boldsymbol{T_1}) \leqslant \boldsymbol{a} \leqslant$ clés$(\boldsymbol{T_2}) \leqslant \boldsymbol{c} \leqslant$ clés$(\boldsymbol{T_3})$

AVL : équilibrage, cas 1a

- hauteur(GG) $=$ hauteur(D) $+ 1$

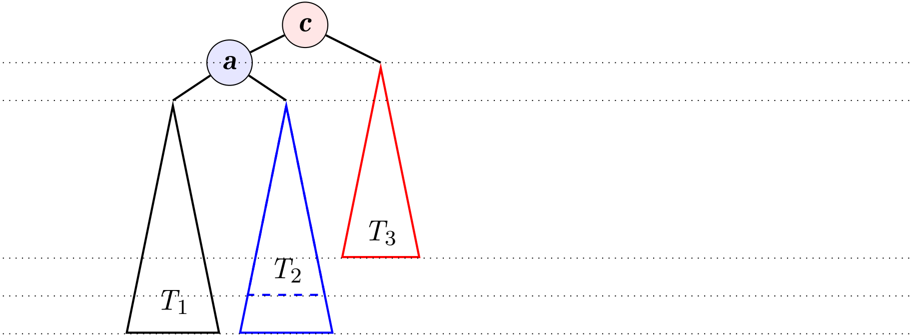


clés$(\boldsymbol{T_1}) \leqslant \boldsymbol{a} \leqslant$ clés$(\boldsymbol{T_2}) \leqslant \boldsymbol{c} \leqslant$ clés$(\boldsymbol{T_3})$

On retrouve la hauteur avant insertion $\rightsquigarrow$ terminé !

AVL : équilibrage, cas 1a

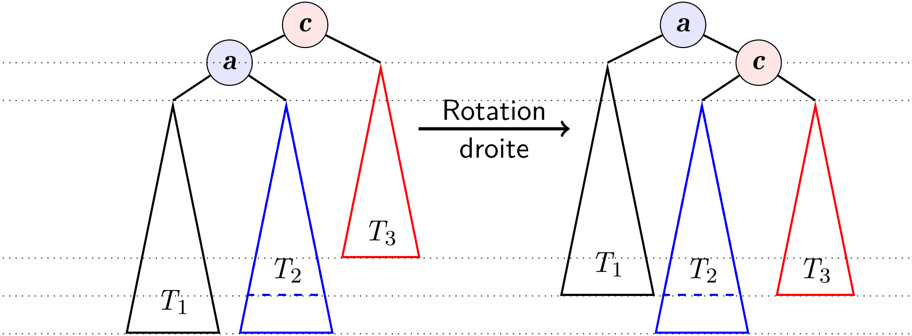
- **Remarque** Fonctionne même si T_2 est plus haut.
  Utile dans le cas de la suppression.



clés$(\boldsymbol{T_1}) \leqslant \boldsymbol{a} \leqslant$ clés$(\boldsymbol{T_2}) \leqslant \boldsymbol{c} \leqslant$ clés$(\boldsymbol{T_3})$

AVL : équilibrage, cas 1a

- **Remarque** Fonctionne même si T_2 est plus haut.
  Utile dans le cas de la suppression.



clés($\boldsymbol{T_1}$) $\leqslant \boldsymbol{a} \leqslant$ clés($\boldsymbol{T_2}$) $\leqslant \boldsymbol{c} \leqslant$ clés($\boldsymbol{T_3}$)

AVL : équilibrage, cas 1b

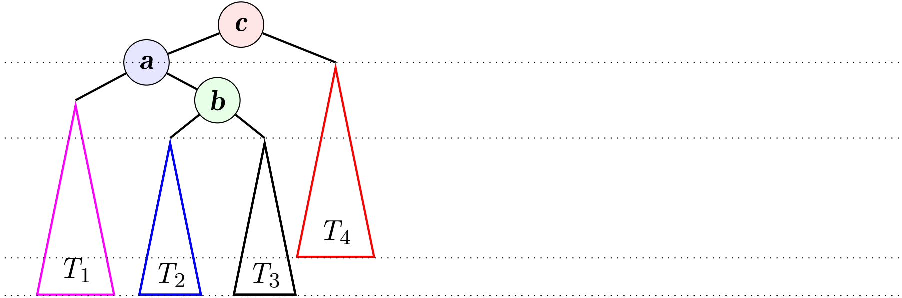
- hauteur(DD) $=$ hauteur(G) $+ 1$

Symétrique du cas 1a) : rotation gauche.

AVL : équilibrage, cas 2a

- Insertion dans le sous-arbre droit du sous-arbre gauche (GD).
  La hauteur de T_2 ou celle de T_3 a augmenté.

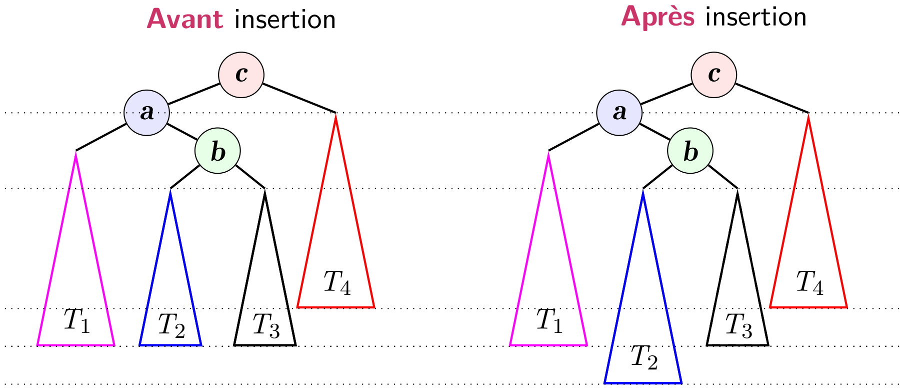
Avant insertion



$$\text{clés}(\boldsymbol{T_1}) \leqslant \boldsymbol{a} \leqslant \text{clés}(\boldsymbol{T_2}) \leqslant \boldsymbol{b} \leqslant \text{clés}(\boldsymbol{T_3}) \leqslant \boldsymbol{c} \leqslant \text{clés}(\boldsymbol{T_4})$$

AVL : équilibrage, cas 2a

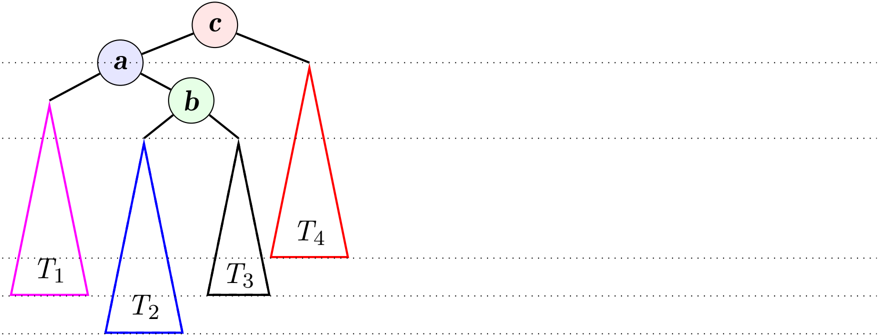
- Insertion dans le sous-arbre droit du sous-arbre gauche (GD).
  La hauteur de T_2 ou celle de T_3 a augmenté.



$$\text{clés}(\boldsymbol{T_1}) \leqslant \boldsymbol{a} \leqslant \text{clés}(\boldsymbol{T_2}) \leqslant \boldsymbol{b} \leqslant \text{clés}(\boldsymbol{T_3}) \leqslant \boldsymbol{c} \leqslant \text{clés}(\boldsymbol{T_4})$$

AVL : équilibrage, cas 2a

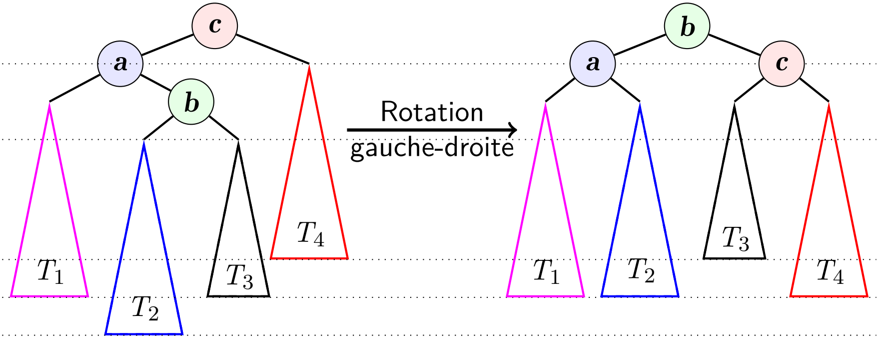
- hauteur(GD) $=$ hauteur(D) $+ 1$



$$\text{clés}(\boldsymbol{T_1}) \leqslant \boldsymbol{a} \leqslant \text{clés}(\boldsymbol{T_2}) \leqslant \boldsymbol{b} \leqslant \text{clés}(\boldsymbol{T_3}) \leqslant \boldsymbol{c} \leqslant \text{clés}(\boldsymbol{T_4})$$

AVL : équilibrage, cas 2a

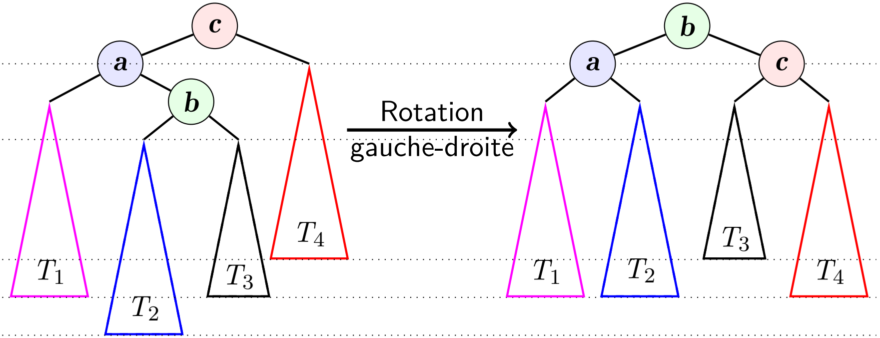
- hauteur(GD) $=$ hauteur(D) $+ 1$



$$\text{clés}(\boldsymbol{T_1}) \leqslant \boldsymbol{a} \leqslant \text{clés}(\boldsymbol{T_2}) \leqslant \boldsymbol{b} \leqslant \text{clés}(\boldsymbol{T_3}) \leqslant \boldsymbol{c} \leqslant \text{clés}(\boldsymbol{T_4})$$

AVL : équilibrage, cas 2a

- hauteur(GD) $=$ hauteur(D) $+ 1$



clés$(\boldsymbol{T_1}) \leqslant \boldsymbol{a} \leqslant$ clés$(\boldsymbol{T_2}) \leqslant \boldsymbol{b} \leqslant$ clés$(\boldsymbol{T_3}) \leqslant \boldsymbol{c} \leqslant$ clés$(\boldsymbol{T_4})$

On retrouve la hauteur avant insertion $\rightsquigarrow$ terminé !

AVL : équilibrage, cas 2b

- hauteur(DG) $=$ hauteur(D) $+ 1$

Symétrique du cas 2a) : rotation droite-gauche.

Insertion dans les AVL : résumé

L'idée est la même que pour l'insertion/suppression dans les tas :

1. On insère en mode ABR en cassant temporairement la propriété AVL.
2. On répare.

Insertion dans les AVL : résumé

L'idée est la même que pour l'insertion/suppression dans les tas :

1. On insère en mode ABR en cassant temporairement la propriété AVL.
2. On répare.

Comment mémoriser les rotations ?

Suppression dans les arbres AVL

La suppression se gère de la même façon que l'insertion.

- Suppression comme dans les arbres binaires de recherche.
- Ré-équilibrage pour retrouver la propriété "AVL".

Un ré-équilibrage peut déséquilibrer un nœud plus haut

Suppression dans les arbres AVL

La suppression se gère de la même façon que l'insertion.

- Suppression comme dans les arbres binaires de recherche.
- Ré-équilibrage pour retrouver la propriété "AVL".

Un ré-équilibrage peut déséquilibrer un nœud plus haut

[$\rightsquigarrow$] peut nécessiter plusieurs équilibrages, au pire jusqu'à la racine.

Arbres AVL : complexité insertion, suppression

La phase d'insertion prend un temps $O(\log(n))$ Chaque ré-équilibrage local demande un temps $O(1)$

- 1 rééquilibrage pour l'insertion.
- Au pire $O(h) = O(\log(n))$ rééquilibrages chacuns coûtant $O(1)$ pour la suppression.

En tout : $O(\log(n))$.

Plan

Les arbres rouges et noirs

Un **arbre rouge et noir** est un ABR tel que :

1. il est **complet**

Les arbres rouges et noirs

Un **arbre rouge et noir** est un ABR tel que :

1. il est **complet** (pas de nœud d'arité 1),

Les arbres rouges et noirs

Un **arbre rouge et noir** est un ABR tel que :

1. il est **complet** (pas de nœud d'arité 1),
2. seuls les nœuds internes portent des valeurs,

Les arbres rouges et noirs

Un **arbre rouge et noir** est un ABR tel que :

1. il est **complet** (pas de nœud d'arité 1),
2. seuls les nœuds internes portent des valeurs,
3. chaque nœud est soit rouge, soit noir,

Les arbres rouges et noirs

Un **arbre rouge et noir** est un ABR tel que :

1. il est **complet** (pas de nœud d'arité 1),
2. seuls les nœuds internes portent des valeurs,
3. chaque nœud est soit rouge, soit noir,
4. la racine est noire,

Les arbres rouges et noirs

Un **arbre rouge et noir** est un ABR tel que :

1. il est **complet** (pas de nœud d'arité 1),
2. seuls les nœuds internes portent des valeurs,
3. chaque nœud est soit rouge, soit noir,
4. la racine est noire,
5. les feuilles sont noires,

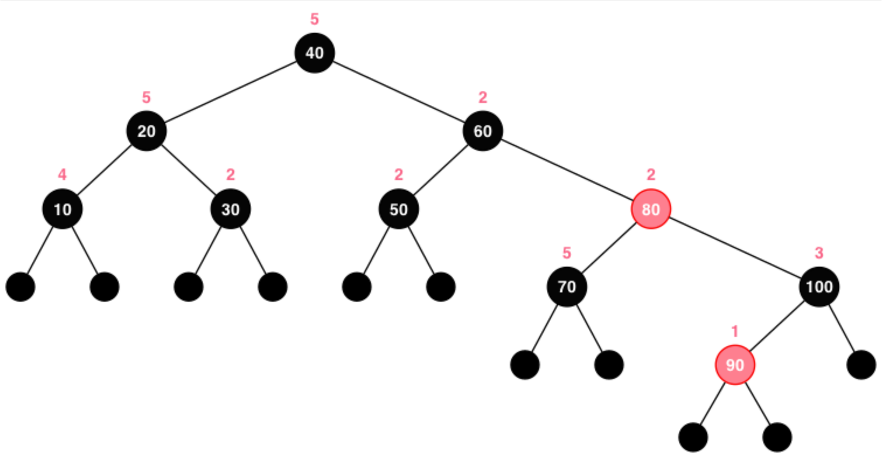
Les arbres rouges et noirs

Un **arbre rouge et noir** est un ABR tel que :

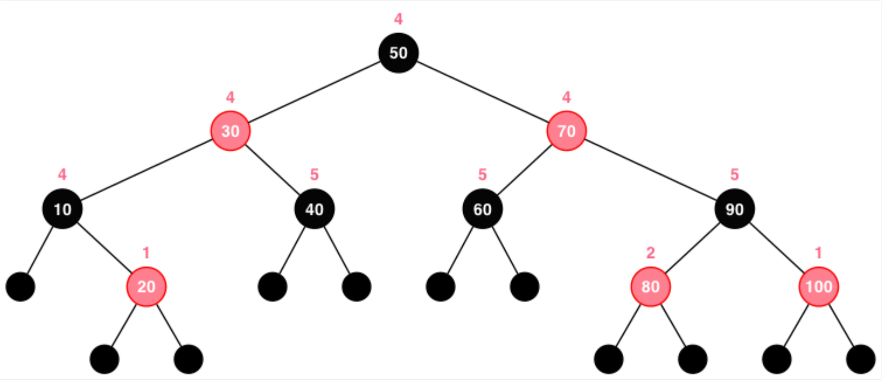
1. il est **complet** (pas de nœud d'arité 1),
2. seuls les nœuds internes portent des valeurs,
3. chaque nœud est soit rouge, soit noir,
4. la racine est noire,
5. les feuilles sont noires,
6. le père d'un nœud rouge est noir.
7. le nombre de nœuds noirs sur chaque branche est **constant**.

Hauteur noire = nombre de nœuds noirs sur chaque branche.

Arbres rouges et noirs : exemple 1



Arbres rouges et noirs : exemple 2



La famille des arbres rouges et noirs est équilibrée

Preuve directe, sans récurrence. Soit t un arbre rouge et noir.

- $H =$ hauteur noire de t, $h =$ hauteur, $n =$ nombre de nœuds.
- L'arbre est complet et les branches ont au moins H nœuds.

  Donc les niveaux 0, 1, ..., $H - 1$ sont **complètement remplis**.

La famille des arbres rouges et noirs est équilibrée

Preuve directe, sans récurrence. Soit t un arbre rouge et noir.

- $H =$ hauteur noire de t, $h =$ hauteur, $n =$ nombre de nœuds.
- L'arbre est complet et les branches ont au moins H nœuds.

  Donc les niveaux 0, 1, ..., $H - 1$ sont **complètement remplis**.
- Donc il y a au minimum $2^H - 1$ nœuds :

  $\boldsymbol{2^H - 1 \leqslant n}$, donc : $\boldsymbol{H \leqslant \log_2(n+1)}$.

La famille des arbres rouges et noirs est équilibrée

Preuve directe, sans récurrence. Soit t un arbre rouge et noir.

- $H =$ hauteur noire de t, $h =$ hauteur, $n =$ nombre de nœuds.
- L'arbre est complet et les branches ont au moins H nœuds.

  Donc les niveaux 0, 1, ..., $H - 1$ sont **complètement remplis**.
- Donc il y a au minimum $2^H - 1$ nœuds :

  $$\boldsymbol{2^H - 1 \leqslant n}, \text{ donc : } \boldsymbol{H \leqslant \log_2(n + 1)}.$$
- Une branche a H nœuds noirs et au maximum $H - 1$ nœuds rouges :

  $\bullet - \bullet - \bullet - \bullet - \cdots - \bullet - \bullet - \bullet$.
- Donc $h \leqslant 2(H - 1)$ et en utilisant l'inégalité ci-dessus :

$$h \leqslant 2(\log_2(n + 1) - 1).$$

Insertion dans un arbre rouge-noir

Un algorithme d'insertion dans un arbre rouge-noir :

- On insère **comme dans un ABR**,
- **Si création** d'un nouveau nœud,
  - il remplace l'une des anciennes feuilles noires,
  - on le colore en **rouge**,
  - ses deux fils sont des feuilles (noires).
- **Problème potentiel**

Insertion dans un arbre rouge-noir

Un algorithme d'insertion dans un arbre rouge-noir :

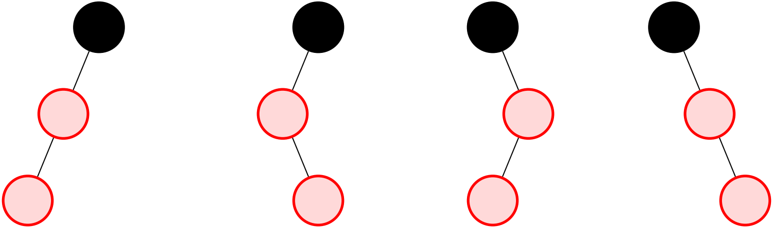
- On insère **comme dans un ABR**,
- **Si création** d'un nouveau nœud,
  - il remplace l'une des anciennes feuilles noires,
  - on le colore en **rouge**,
  - ses deux fils sont des feuilles (noires).

- **Problème potentiel**
  - l'arbre obtenu peut avoir **2 nœuds rouges père-fils**.

Insertion dans un arbre rouge-noir

Réorganisation de l'arbre pour

- éviter 2 nœuds rouges consécutifs.
- conserver les autres propriétés des arbres rouges et noirs.
- Après insertion : 4 cas possibles avec $\mathbf{a} < \mathbf{b} < \mathbf{c}$:

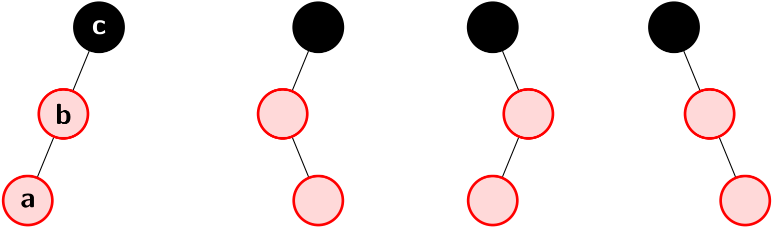


Cas 1a **Cas 2a** **Cas 2b** **Cas 1b**

Insertion dans un arbre rouge-noir

Réorganisation de l'arbre pour

- éviter 2 nœuds rouges consécutifs.
- conserver les autres propriétés des arbres rouges et noirs.
- Après insertion : 4 cas possibles avec **a** $<$ **b** $<$ **c** :

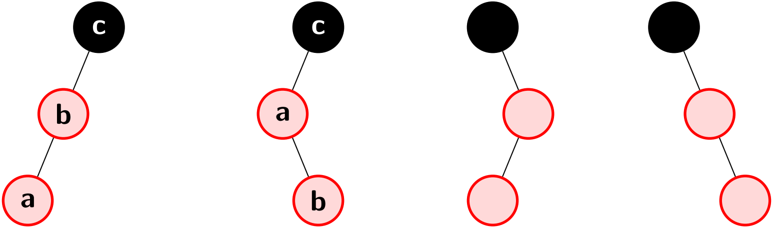


Cas 1a **Cas 2a** **Cas 2b** **Cas 1b**

Insertion dans un arbre rouge-noir

Réorganisation de l'arbre pour

- éviter 2 nœuds rouges consécutifs.
- conserver les autres propriétés des arbres rouges et noirs.
- Après insertion : 4 cas possibles avec **a** $<$ **b** $<$ **c** :

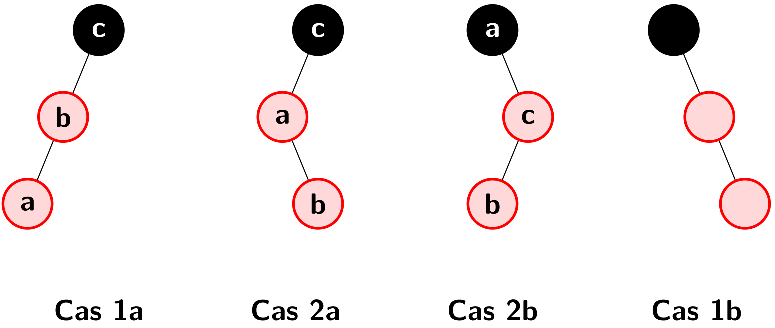


Cas 1a **Cas 2a** **Cas 2b** **Cas 1b**

Insertion dans un arbre rouge-noir

Réorganisation de l'arbre pour

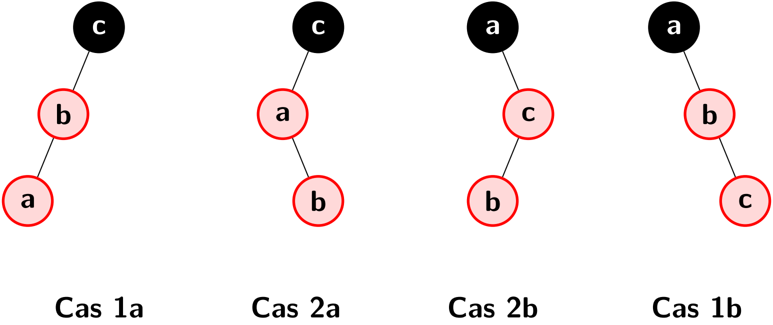
- éviter 2 nœuds rouges consécutifs.
- conserver les autres propriétés des arbres rouges et noirs.
- Après insertion : 4 cas possibles avec $\mathbf{a} < \mathbf{b} < \mathbf{c}$:



Insertion dans un arbre rouge-noir

Réorganisation de l'arbre pour

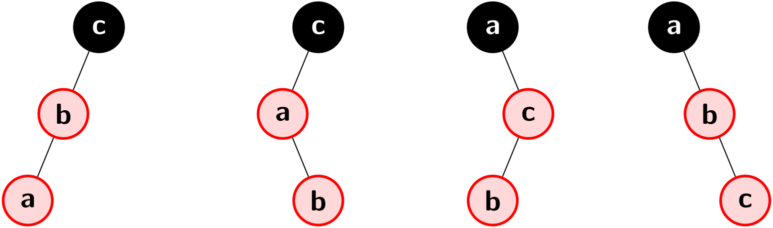
- éviter 2 nœuds rouges consécutifs.
- conserver les autres propriétés des arbres rouges et noirs.
- Après insertion : 4 cas possibles avec **a** $<$ **b** $<$ **c** :



Insertion dans un arbre rouge-noir

Réorganisation de l'arbre pour

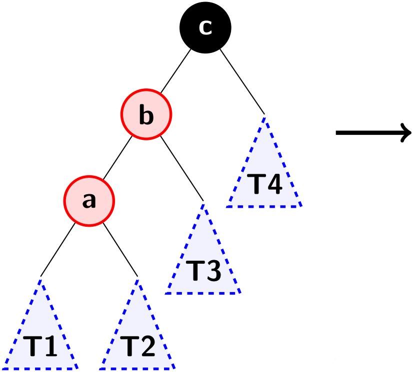
- éviter 2 nœuds rouges consécutifs.
- conserver les autres propriétés des arbres rouges et noirs.
- Après insertion : 4 cas possibles avec $\mathbf{a} < \mathbf{b} < \mathbf{c}$:



Cas 1a **Cas 2a** **Cas 2b** **Cas 1b**

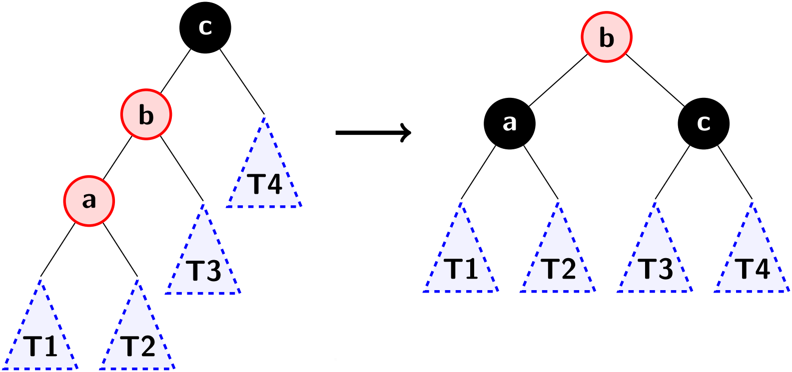
On applique une **transformation** dans chacun des 4 cas.

Insertion dans un arbre rouge-noir : cas 1



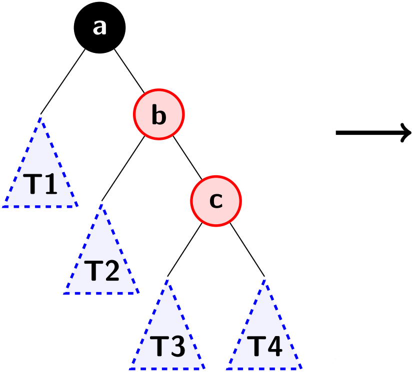
$$\mathbf{cles(T1) < a < cles(T2) < b < cles(T3) < c < cles(T4)}$$

Insertion dans un arbre rouge-noir : cas 1



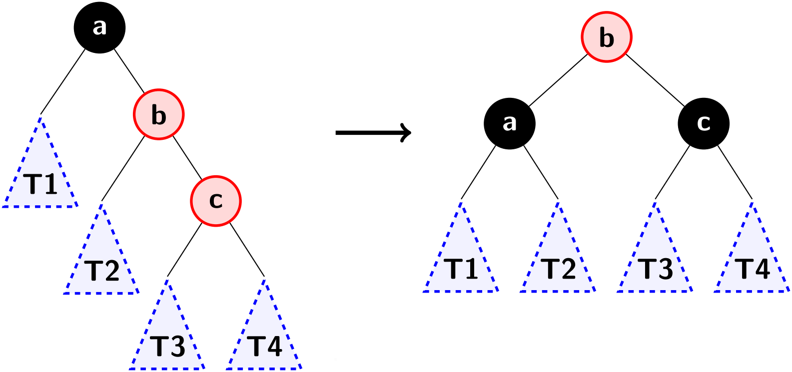
$$\mathbf{cles(T1) < a < cles(T2) < b < cles(T3) < c < cles(T4)}$$

Insertion dans un arbre rouge-noir : cas 1b



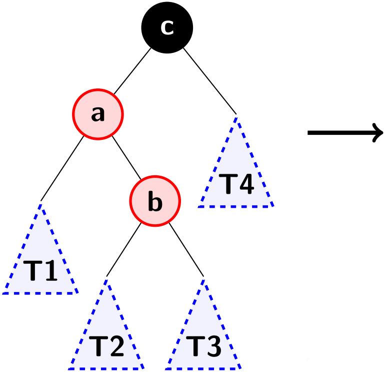
$$\textbf{cles(T1)} < \textbf{a} < \textbf{cles(T2)} < \textbf{b} < \textbf{cles(T3)} < \textbf{c} < \textbf{cles(T4)}$$

Insertion dans un arbre rouge-noir : cas 1b



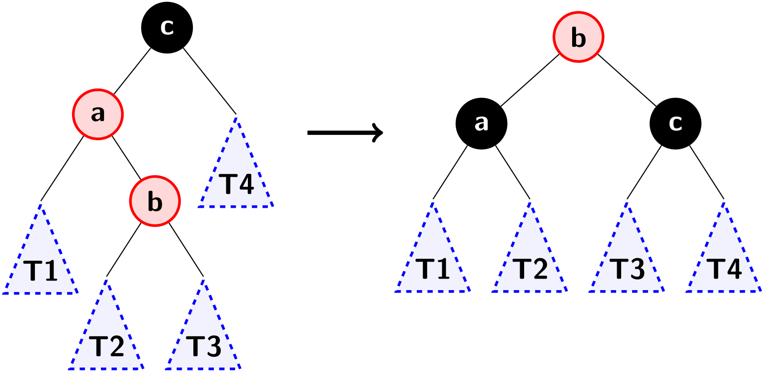
cles(T1) < a < cles(T2) < b < cles(T3) < c < cles(T4)

Insertion dans un arbre rouge-noir : cas 2a



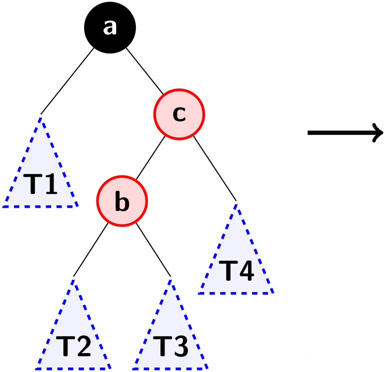
$$\textbf{cles(T1)} < \textbf{a} < \textbf{cles(T2)} < \textbf{b} < \textbf{cles(T3)} < \textbf{c} < \textbf{cles(T4)}$$

Insertion dans un arbre rouge-noir : cas 2a



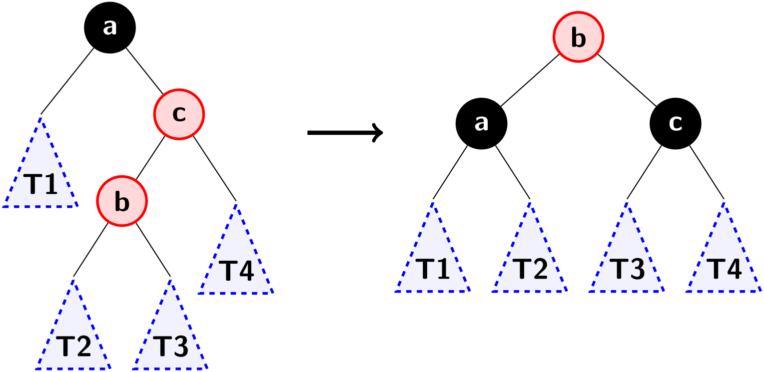
cles(T1) < a < cles(T2) < b < cles(T3) < c < cles(T4)

Insertion dans un arbre rouge-noir : cas 2b



$$\textbf{cles(T1)} < \textbf{a} < \textbf{cles(T2)} < \textbf{b} < \textbf{cles(T3)} < \textbf{c} < \textbf{cles(T4)}$$

Insertion dans un arbre rouge-noir : cas 2b

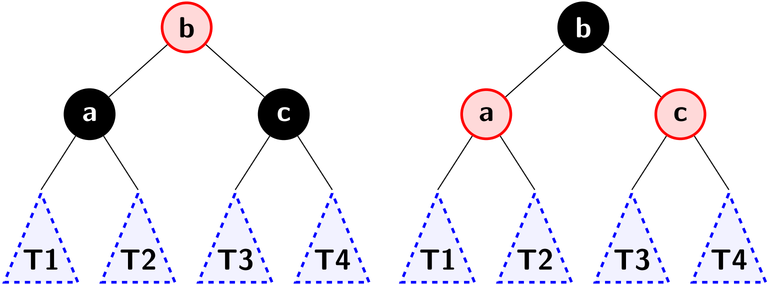


cles(T1) < a < cles(T2) < b < cles(T3) < c < cles(T4)

Question...

Au lieu de

Peut-on choisir



????

Insertion et rééquilibrage

- La transformation crée un nœud rouge à la racine du sous-arbre sur lequel on travaille.
- $\Rightarrow$ Risque de créer à nouveau 2 nœuds rouges consécutifs.
- $\Rightarrow$ On doit ré-équilibrer récursivement.
- Si on arrive à la racine, on peut la colorier en noir.
  Augmente la hauteur noire de 1 sans changer le fait qu'elle est constante.

Suppression dans les arbres rouges-noirs

- Plus compliquée.
- On supprime d'abord comme dans un ABR normal.
- Problème quand on supprime un nœud noir : le nombre de nœuds noirs est maintenant inférieur.
- **Une solution** :
  - créer un nœud « double noir »,
  - le faire remonter, potentiellement jusqu'à la racine, par rotations.
  - le transformer en nœud noir s'il est à la racine.
  - la remontée du nœud double noir peut conduire à créer un nœud noir négatif.

Arbres rouges et noirs : complexité insertion, suppression

- La phase d'insertion sans ré-équilibrage prend un temps $O(\log(n))$
- Chaque ré-équilibrage local demande un temps $O(1)$
- Au pire $O(h) = O(\log(n))$ rééquilibrages chacuns coûtant $O(1)$ pour la suppression.
- **En tout** : $O(\log(n))$.