Les arbres binaires de recherche équilibrés

- Introduction
- Arbres binaires de recherche
- Arbres AVL
- Arbres rouge-noir
- Performances
- Conclusion

Arbres binaires de recherche

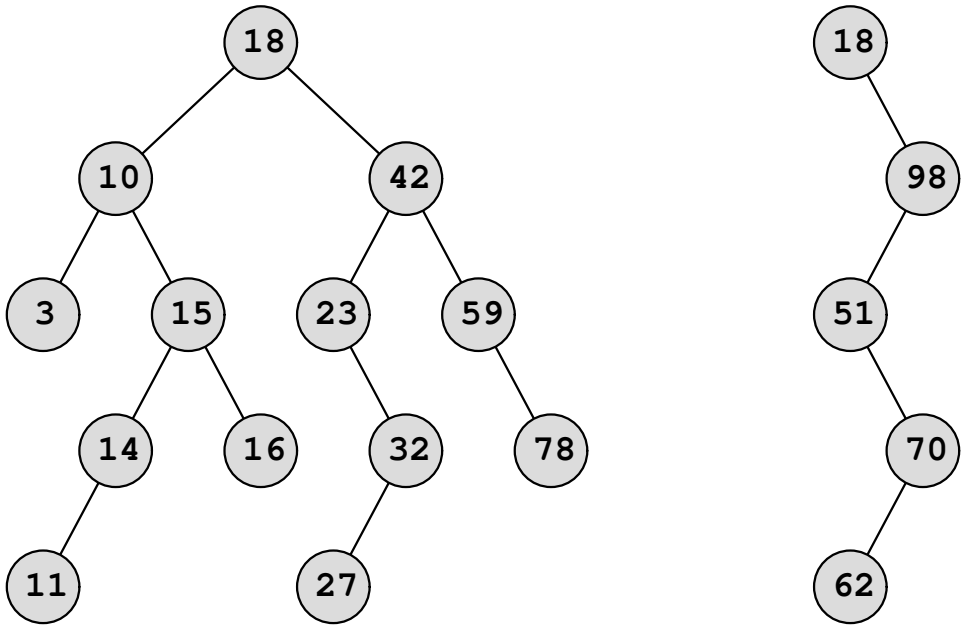
Découverts vers la fin des années 1950.

Définition 1 (Arbre binaire). Soit E un ensemble. Un *arbre binaire* est :

- soit l'arbre vide $\varnothing$;
- soit un *nœud* $A(g, r, d)$, où g désigne le sous-arbre gauche, d le sous-arbre droit, et $r \in E$ les données satellites stockées dans le nœud.

Définition 2 (Arbre binaire de recherche). Un arbre binaire est *de recherche* lorsque, si x est un nœud de l'arbre, et y un nœud du sous-arbre gauche (resp. droit) de x, on a $y < x$ (resp. $x < y$).

Exemples d'arbres binaires de recherche



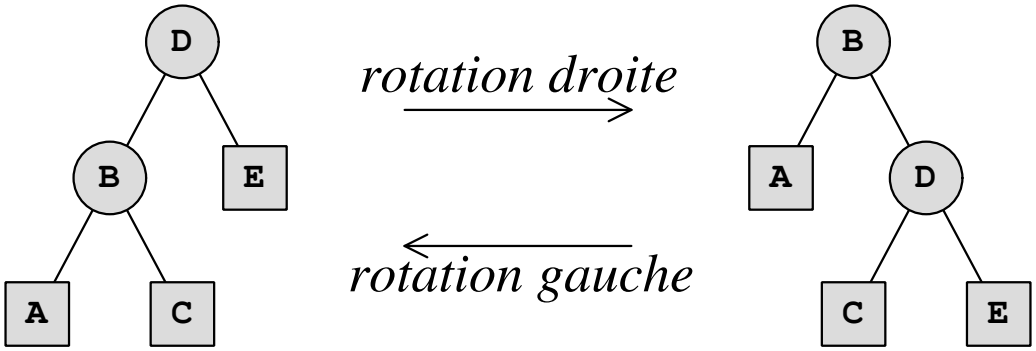
Hauteur d'un arbre binaire

Proposition 1. *Soit un arbre binaire non vide de hauteur h et possédant n nœuds. On a :*

$$\lfloor \log_2 n \rfloor \leqslant h \leqslant n - 1.$$

Ces bornes sont optimales.

Rotations



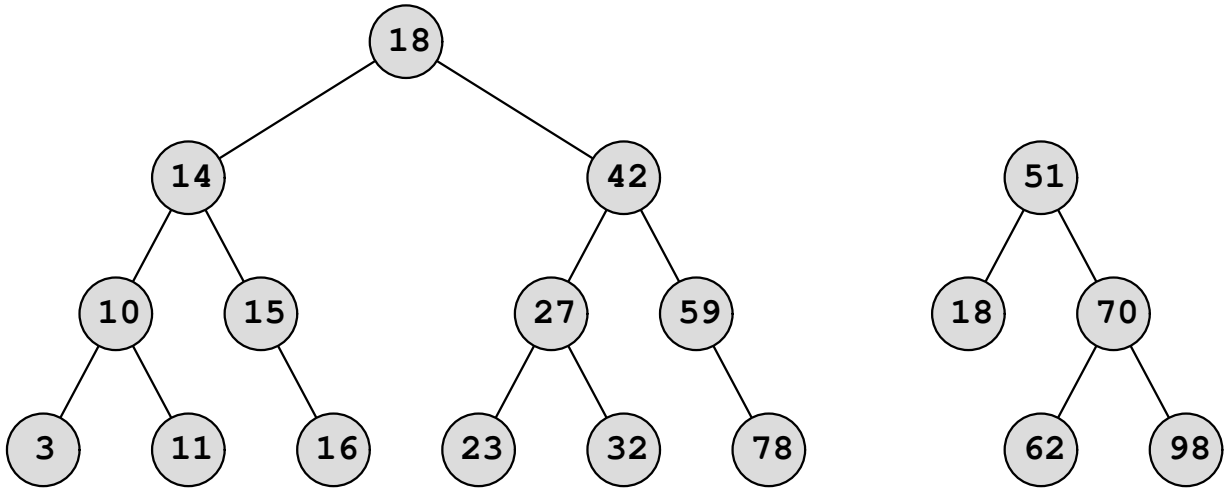
Proposition 2. *Les rotations préservent la propriété d'arbre binaire de recherche.*

Arbres AVL

Introduits pour la première fois par Adel'son-Vel'skiĭ et Landis en 1962.

Définition 3. Un arbre binaire de recherche est un *arbre AVL* si, pour n'importe lequel de ses nœuds, la différence de hauteur entre ses deux fils diffère d'au plus un.

Exemples d'arbres AVL



Hauteur d'un arbre AVL

Proposition 3. *Soit un arbre AVL de hauteur* h *et possédant* n *nœuds. On a :*

$$h \leqslant \frac{3}{2}\log_2(n+1).$$

Hauteur d'un arbre AVL

Démonstration. Soit u_h le nombre minimal de nœuds d'un arbre de hauteur $h \geqslant 0$. On a $u_0 = 1, u_1 = 2$, et :

$$\forall h \in \mathbb{N}, \quad u_{h+2} = u_h + u_{h+1} + 1.$$

Soit, après résolution :

$$u_h = A\alpha^h + B\beta^h - 1,$$

avec :

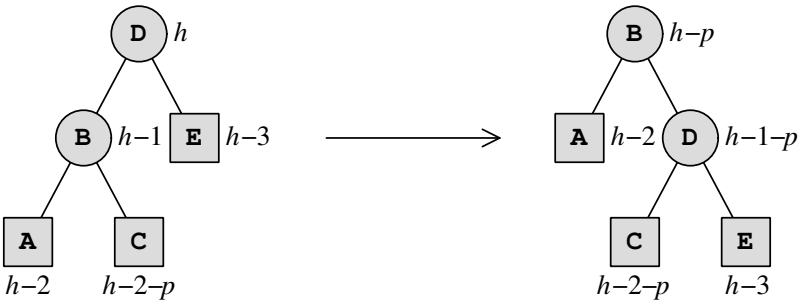
$$A = \frac{5 + 2\sqrt{5}}{5} \approx 1,89, \qquad B = \frac{5 - 2\sqrt{5}}{5} \approx 0,11,$$
$$\alpha = \frac{1 + \sqrt{5}}{2} \approx 1,62, \qquad \beta = \frac{1 - \sqrt{5}}{2} \approx -0,62.$$

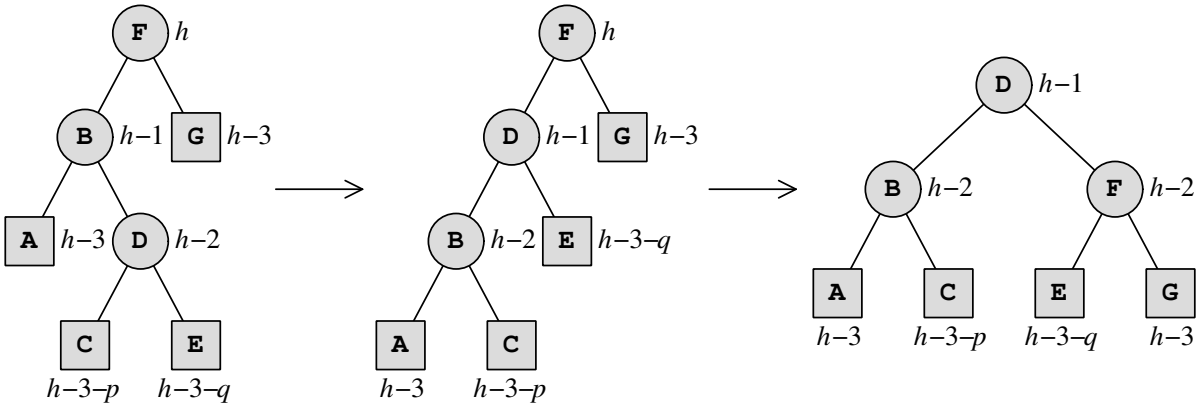
On a donc :

$$\begin{aligned} n \geqslant u_h &> A\alpha^h - 2, \\ n + 2 &> A\alpha^h, \\ h &< \log_\alpha(n+1) + \log_\alpha\left(\frac{1}{A} \cdot \frac{n+2}{n+1}\right), \\ h &< \log_\alpha(n+1) = \frac{\ln 2}{\ln \alpha} \log_2(n+1), \\ h &< \frac{3}{2}\log_2(n+1). \end{aligned}$$

L'inégalité est vraie pour $h = -1$. □

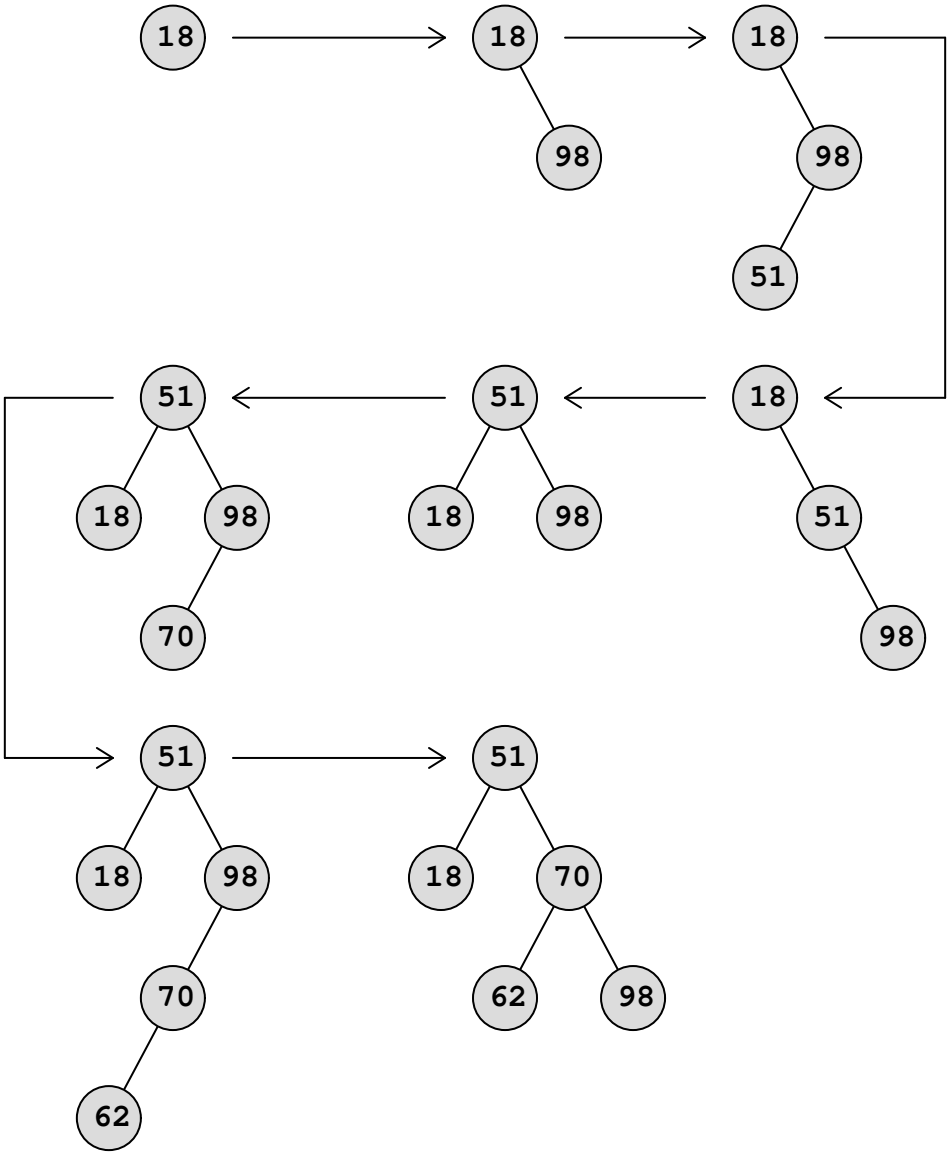
Rééquilibrage d'un arbre AVL





Exemple de construction d'un arbre AVL

Détail de l'insertion de 18, 98, 51, 70 et 62 dans un arbre vide.



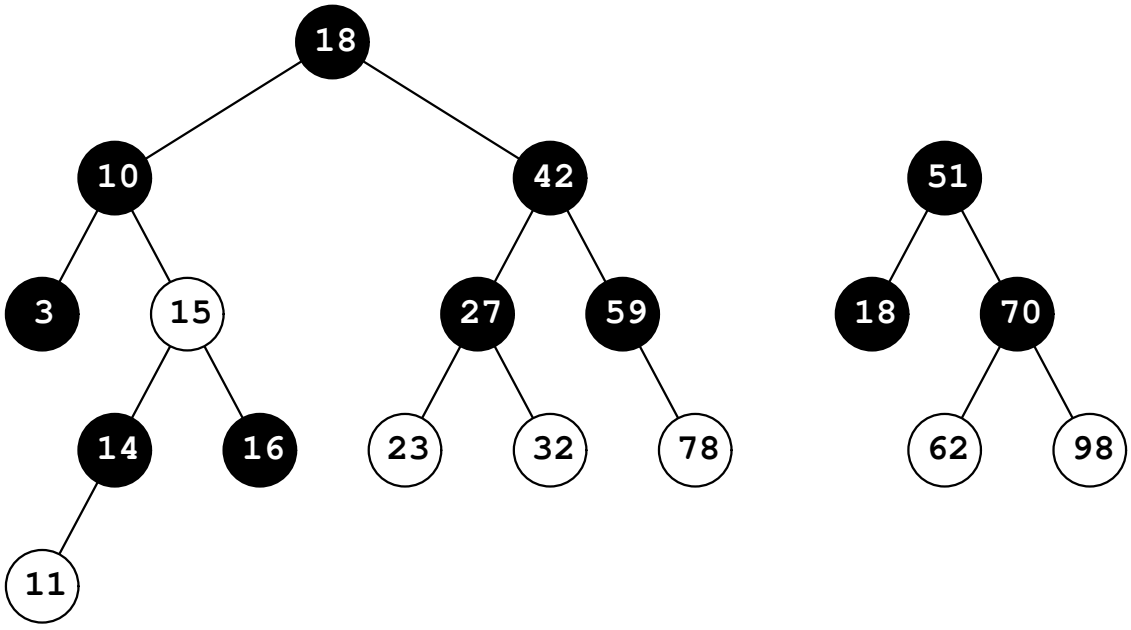
Arbres rouge-noir

Inventés par Bayer en 1972. Étudiés en détail par Guibas et Sedgewick en 1978.

Définition 4. Un arbre binaire de recherche est un *arbre rouge-noir* s'il vérifie les propriétés suivantes :

1. chaque nœud est soit rouge, soit noir ;
2. la racine est noire ;
3. chaque sous-arbre vide est noir ;
4. si un nœud est rouge, alors ses deux enfants sont noirs ;
5. pour chaque nœud, tous les chemins reliant le nœud à une feuille contiennent le même nombre de nœuds noirs (ce nombre est appelé *hauteur noire*).

Exemples d'arbres rouge-noir



Hauteur d'un arbre rouge-noir

Proposition 4. *Soit un arbre rouge-noir de hauteur h et possédant n nœuds. On a :*

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

Hauteur d'un arbre rouge-noir

Démonstration. Lemme : un sous-arbre de hauteur h et de hauteur noire ω possède au moins $2^\omega - 1$ nœuds.

Pour $h = -1$ ou $h = 0$, c'est évident.

Soit $A(g, r, d)$ un arbre rouge-noir de hauteur $h + 1$. Alors $\omega(g) \geqslant \omega - 1$, et $\omega(d) \geqslant \omega - 1$. On a alors :

$$\begin{aligned} n &\geqslant \left(2^{\omega-1} - 1\right) + \left(2^{\omega-1} - 1\right) + 1, \\ &\geqslant 2^\omega - 1. \end{aligned}$$

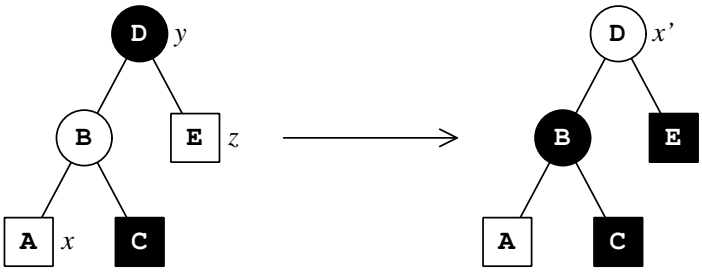
Le lemme est ainsi démontré au rang $h + 1$.

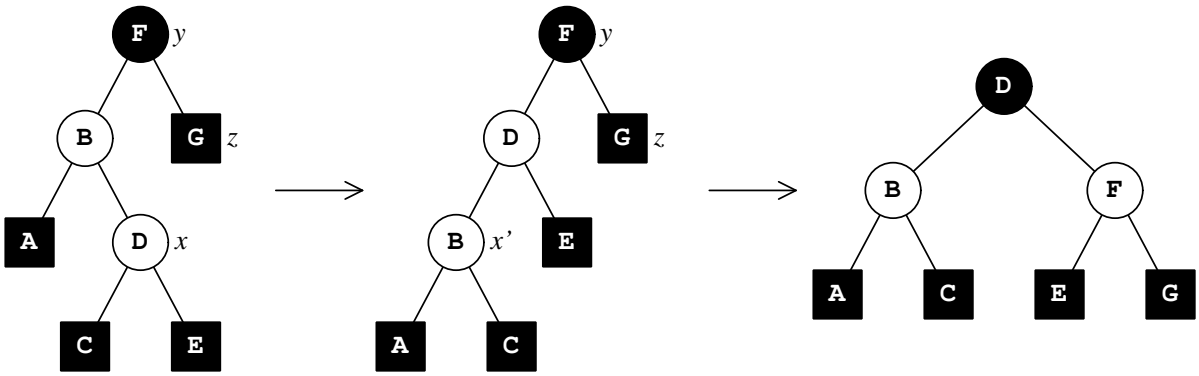
Un arbre rouge-noir non vide vérifie $\omega \geqslant h/2$. En appliquant le lemme, il vient : $n \geqslant 2^{h/2} - 1$, d'où :

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

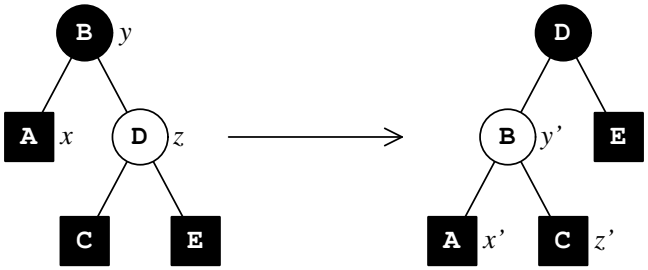
L'inégalité est vraie pour $n = 0$. □

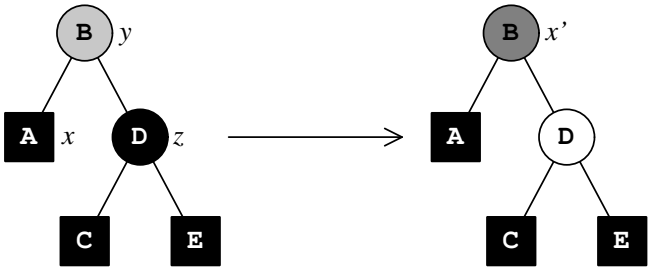
Insertion dans un arbre rouge-noir

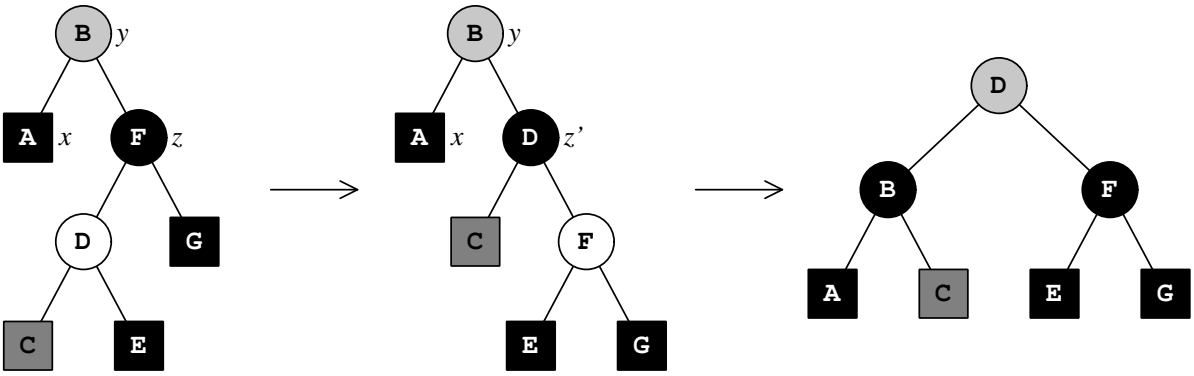




Suppression d'un arbre rouge-noir

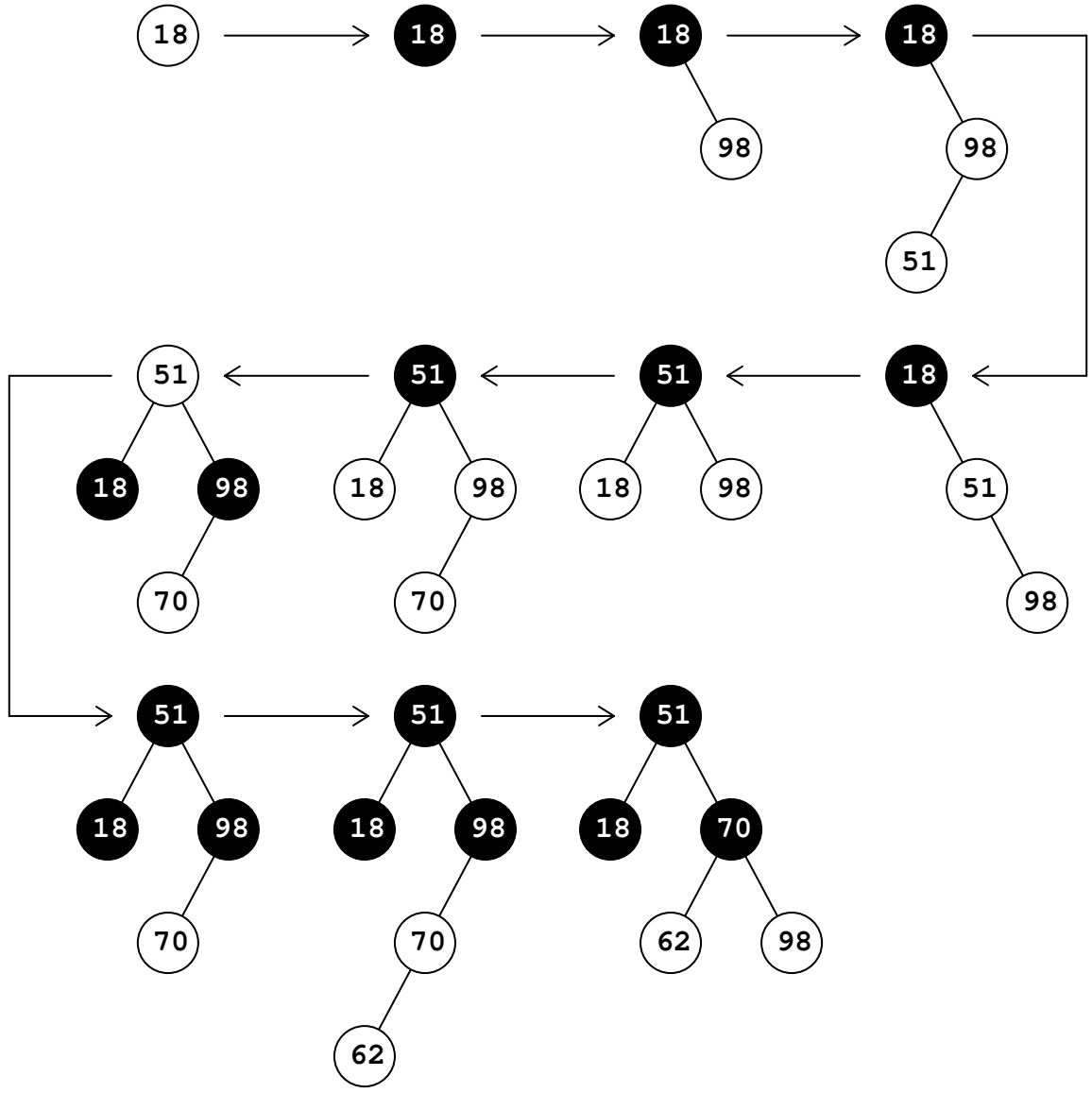






Exemple de construction d'un arbre rouge-noir

Détail de l'insertion de 18, 98, 51, 70 et 62 dans un arbre vide.



Arbres construits aléatoirement

Algorithme	Test	h	τ_0 (s)	τ_1 (s)	τ_2 (s)	τ_3 (s)
Naïf	1	41	19,73	0,08	0,04	0,04
AVL	1	18	19,71	0,08	0,05	0,08
Rouge-Noir	1	19	20,09	0,08	0,05	0,06
Naïf	2	45	47,02	0,20	0,13	0,07
AVL	2	19	46,03	0,20	0,14	0,17
Rouge-Noir	2	20	47,05	0,20	0,14	0,13
Naïf	3	42	109,81	0,44	0,36	0,15
AVL	3	20	107,30	0,47	0,26	0,38
Rouge-Noir	3	21	108,79	0,49	0,33	0,27

Test 1 : $m = 1\,000\,000$, $n = 64\,000$ et $p = 4\,000$
Test 2 : $m = 2\,000\,000$, $n = 128\,000$ et $p = 8\,000$
Test 3 : $m = 4\,000\,000$, $n = 256\,000$ et $p = 16\,000$

n : nombre de nœuds de l'arbre

h : hauteur de l'arbre

τ_0 : insertion/recherche de m clés aléatoires

τ_1 : suppression de p clés aléatoires

τ_2 : insertion de p clés aléatoires

τ_3 : suppression des n clés dans l'ordre croissant

Test d'un des cas les pires

Algorithme	Test	h	$\frac{h}{\log_2 n}$	τ_0 (s)	$\frac{\tau_0}{n \log_2 n}$ (μs)
Naïf	1	31 999	2 138	614	1 383
AVL	1	14	0,9	0,06	0,13
AVL	2	18	0,9	1,31	0,14
AVL	3	20	1,0	5,51	0,14
Rouge-Noir	1	26	1,7	0,08	0,17
Rouge-Noir	2	34	1,8	1,79	0,19
Rouge-Noir	3	38	1,8	9,76	0,24

Test 1 : $n = 32\,000$
Test 2 : $n = 512\,000$
Test 3 : $n = 2\,000\,000$

n : nombre de nœuds de l'arbre

h : hauteur de l'arbre

τ_0 : insertion des n clés dans l'ordre croissant