TP n°22 - Équilibrage des ABR Rouge/Noir - Corrigé

I. Arbres bicolores rouge-noir

Définition 1 (Arbre bicolore)

Un **arbre bicolore rouge-noir** est
- soit l'arbre vide E
- soit un arbre binaire t pour lequel chaque nœud est associé soit à la couleur **rouge**, soit à la couleur **noire**, et qui vérifie les propriétés suivantes :
  - **(P1)** t est un ABR
  - **(P2)** La **racine** est **noire**.
  - **(P3)** Un nœud rouge ne peut pas avoir d'enfant rouge.
  - **(P4)** Pour un nœud donné, tous les chemins de ce nœud à un sous arbre vide comportent le même nombre de nœuds noirs.

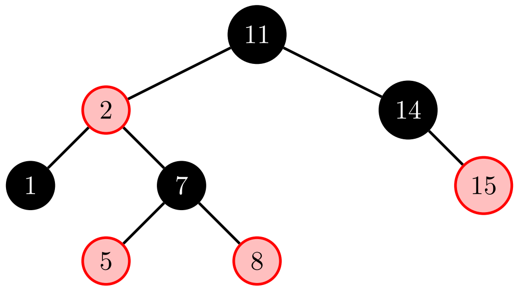


FIGURE 1 – Exemple d'arbre rouge-noir

Exercice 1 (Étude théorique : équilibrage des arbres bicolores).

1. Indiquer pourquoi chacun des arbres de la figure 2 n'est pas un arbre rouge-noir.
2. On considère le nombre de nœuds noirs d'un nœud vers n'importe quelle feuille, notion bien définie d'après **(P4)**.
   Quelle est ce nombre de nœuds noirs $b(t)$ de l'arbre rouge-noir de l'exemple 1.
3. Justifier que l'on peut remplacer la propriété **(P4)** par : **tout chemin de la racine à une feuille comporte le même nombre de nœuds noirs**.
4. Soit t un arbre vérifiant $(P3)$ et $(P4)$. On note $b(t)$ le nombre de nœuds noirs de la racine à une feuille de t. Cette valeur $b(t)$ sera appelée **hauteur noire** de l'arbre t. Montrer que l'on a : $2^{b(t)} \leq n(t)+1$, où $n(t)$ désigne le nombre de nœuds de l'arbre t.
5. Montrer que, pour tout arbre t vérifiant $(P3)$ et $(P4)$, la hauteur $h(t)$ de cet arbre vérifie : $h(t) \leq 2b(t)$
6. Montrer que les arbres bicolores forment une famille d'arbres équilibrés.
7. Comparer la famille des arbres rouge-noir et celle des arbres AVL en terme d'équilibrage, de complexité pour la mise en œuvre algorithmique...

Corrigé de l'exercice 1. [Retour à l'énoncé]

1.
   - L'arbre (a) ne respecte pas la propriété $(P4)$ car le chemin de 11 a 15 possède 2 nœuds noirs, alors que les autres chemins de la racine vers les feuilles 1, 5 et 8 possèdent 3 nœuds noirs.
   - L'arbre (b) a une racine rouge, et ne respecte donc pas la propriété $(P2)$.

— Le nœud d'étiquette 2 de l'arbre (c) est rouge est à un fils, d'étiquette 7, qui est rouge : la propriété $(P3)$ n'est donc pas respectée

— L'arbre (d) n'est pas un ABR car le nœud d'étiquette 0 est mal positionné. Il ne respecte donc pas la propriété $(p1)$

2. Pour l'arbre de la figure 1, on a $b(t) = 2$

3. Il s'agit de montrer l'équivalence entre

$(P4)$: Pour un nœud donné, tous les chemins de ce nœud à une feuille comportent le même nombre de nœuds noirs

et

$(P4')$: Tout chemin de la racine à une feuille comporte le même nombre de nœuds noirs

$(P4) \Rightarrow (P4')$ trivialement, en choisissant comme *nœud donné* la racine de l'arbre.

$(P4') \Rightarrow (P4)$ peut se montrer par induction structurelle. On peut également le montrer par l'absurde.

$\mathcal{P}(t)$: Tout chemin de la racine à n'importe quel sous-arbre vide de t comporte le même nombre de nœuds noirs $\Rightarrow$ Pour tout nœud de t, tous les chemins de ce nœud à n'importe quel sous-arbre vide comportent le même nombre de nœuds noirs.

Cas de base : pour $t = E$, il n'y a aucun nœud dans l'arbre, donc la vérification est triviale.

Induction : On fixe $t \in \mathcal{T} \setminus \{E\}$ et on suppose $(P4')$ pour cet arbre. Par ailleurs, supposons la propriété $\mathcal{P}(t')$ vraie $\forall\, t' \prec t$ (hypothèse d'induction).

Comme $t \neq E$, t a été construit avec l'unique constructeur $t = N(c, \ell, x, r)$.
Comme $(P4')$ est supposée vraie pour t, on peut noter $b(t)$ le nombre de nœuds noirs entre sa racine et n'importe quel sous-arbre vide. Deux cas se présentent.

Si la racine de t est noire ($c = B$), alors $b(\ell) = b(r) = b(t) - 1$.
Le nombre de nœuds noirs entre la racine de ℓ et n'importe lequel de ses sous-arbres vides est constant, car on sait qu'il est égal à $b(t) - 1$ car la racine de t, juste au-dessus, est noire. Comme $\ell \prec t$, on peut lui appliquer l'hypothèse d'induction et on a donc que la partie droite de l'implication est vraie. Cela signifie que, pour n'importe quel noeud de ℓ, tous les chemins de ce nœud à un sous-arbre vide comportent le même nombre de nœuds noirs.
On fait de même pour r : le nombre de nœuds noirs entre la racine de r et n'importe laquelle de ses sous-arbres vides est constant égal à $b(t) - 1$. Comme $r \prec t$, on peut lui appliquer l'hypothèse d'induction et on a donc que, pour un nœud donné de r, tous les chemins de ce nœud à un sous-arbre comportent le même nombre de nœuds noirs.
On a donc montré que pour tous les nœuds du sous-arbre gauche et du sous arbre droit, tous les chemins de ce nœud à l'un de leurs sous-arbres nodes comportent le même nombre de nœuds noirs. C'est vrai aussi pour le nœud racine de t par hypothèse. Ainsi, pour n'importe quel nœud de t (la racine, tous les nœuds de gauche, tous les nœuds de droite), tous les chemins de ce nœud à un sous-arbre vide comportent le même nombre de nœuds noirs et on a montré que $(P4)$ est alors vraie pour t

Si sa racine est rouge ($c = R$), $b(\ell) = b(r) = b(t)$ et on refait exactement le même raisonnement.

Conclusion : par principe d'induction, l'implication $(P4') \Rightarrow (P4)$ a donc été démontrée pour tout arbre t

4. On démontre la propriété suivante par induction structurelle pour t vérifiant $(P3)$ et $(P4)$:

$$\mathcal{P}(t) :\ 2^{b(t)} \leq n(t) + 1$$

Cas de base : Pour $t = E$, $n(t) = 0$ et $b(t) = 0$, la propriété est donc vraie car $2^0 = 1 \leq 0 + 1$.

Induction : On fixe $t \in \mathcal{T} \setminus \{E\}$ et on suppose que $\mathcal{P}(t')$ est vrai pour tout $t' \prec t$ vérifiant $(P3)$ et $(P4)$: Comme $t \neq E$, $t = N(c, \ell, x, r)$.
Comme t vérifie $(P4)$, on a forcément $b(\ell) = b(r)$ en appliquant la propriété $(P4)$ aux nœuds gauche et droit de la racine. Comme $\ell \prec t$ et comme ℓ vérifie $(P3)$ et $(P4)$, on a, par hypothèse d'induction $2^{b(\ell)} \leq n(\ell) + 1$. De même, comme $r \prec t$ et comme r vérifie $(P3)$ et $(P4)$, on a, par

hypothèse d'induction $2^{b(r)} \leq n(r)+1$.
Par ailleurs, par définition :

$$n(t) = 1 + n(\ell) + n(r)$$

On procède par disjonction de cas.

Supposons que la racine de t est noire. Alors $b(t) = 1 + b(r) = 1 + b(l)$ et

$$n(t) = 1+n(\ell)+n(r) \geq 1+2^{b(l)}-1+2^{b(r)}-1 = 2^{b(t)-1}+2^{b(t)-1}-1 = 2\times 2^{b(t)-1}-1 = 2^{b(t)}-1$$

et $\mathcal{P}(t)$ est vérifiée.

Supposons que la racine de t est rouge. Si t ne contient qu'un seul nœud-racine, alors $n(t) = 1$, $b(t) = 0$ et l'inégalité est vraie.
Sinon, $b(\ell) = b(r) = b(t)$ et :

$$n(t) = 1 + n(\ell) + n(r) \geq 1 + 2^{b(l)} - 1 + 2^{b(r)} - 1 = 2^{b(t)} + 2^{b(t)} - 1 = 2 \times 2^{b(t)} - 1 > 2^{b(t)} - 1$$

Dans les deux cas (racine rouge ou noire), on a donc montré que $\mathcal{P}(t)$ est vraie sous hypothèse d'induction

Conclusion : Par principe d'induction, $\mathcal{P}(t)$ est vraie pour tout arbre t vérifiant $(P3)$ et $(P4)$.

5. On démontre la propriété suivante par induction structurelle pour t vérifiant $(P3)$ et $(P4)$:

$$\mathcal{P}(t) : h(t) \leq 2b(t)$$

Cas de base : Pour $t = E$, $h(t) = -1$ et $b(t) = 0$, la propriété est donc vraie car $-1 \leq 2 \times 0$.

Induction : On fixe $t \in \mathcal{T} \setminus \{E\}$ et on suppose que $\mathcal{P}(t')$ est vrai pour tout $t' \prec t$ vérifiant $(P3)$ et $(P4)$. Comme $t \neq E$, $t = N(c, \ell, x, r)$.
Comme t vérifie $(P4)$, on a forcément $b(\ell) = b(r)$ en appliquant la propriété $(P4)$ aux nœuds gauche et droit de la racine. Comme $\ell \prec t$ et comme ℓ vérifie lui aussi $(P3)$ et $(P4)$, on a, par hypothèse d'induction $h(\ell) \leq 2b(\ell)$. De même, comme $r \prec t$ et comme r vérifie $(P3)$ et $(P4)$, on a, par hypothèse d'induction $h(r) \leq 2b(r)$.
Par ailleurs, par définition :

$$h(t) = 1 + \max(h(\ell), h(r))$$

On procède par disjonction de cas.

Supposons que la racine de t est noire. Alors $b(t) = 1 + b(r) = 1 + b(l)$ et

$$h(t) = 1 + \max(h(\ell), h(r))) \leq 1 + 2(b(t) - 1) = 2b(t) - 1 \leq 2b(t)$$

Supposons que la racine de t est rouge. Si t ne contient qu'un seul nœud-racine, alors $h(t) = 0$, $b(t) = 0$ et l'inégalité est vraie.
Sinon :

$$h(t) = 1 + \max(h(\ell), h(r))) \leq 1 + 2b(t)$$

Cela en marche pas... il nous faut trouver une majoration plus fine pour conclure. On descend donc encore dans l'arbre.
Notons $\ell\ell$ et ℓr les deux petits enfants gauche et $r\ell$ et rr les deux petits enfants droits. Alors

$$h(t) = 2 + \max(h(\ell\ell), h(\ell r), h(r\ell), h(rr))$$

Comme la racine est rouge, les deux fils gauche et droit sont forcément noirs. et on a donc :

$$b(\ell\ell) = b(\ell r) = b(r\ell) = b(rr) = b(t) - 1$$

Par ailleurs, comme ces 4 sous-arbres sont strictement inférieurs à t pour l'ordre structurel, on peut leur appliquer l'hypothèse d'induction. On a donc finalement :

$$h(t) = 2 + \max(h(\ell\ell), h(\ell r), h(r\ell), h(rr)) \leq 2 + 2 \times (b(t) - 1) = 2b(t)$$

Dans les deux cas (racine rouge ou noire), on a donc montré que $\mathcal{P}(t)$ est vraie sous hypothèse d'induction

Conclusion : Par principe d'induction, $\mathcal{P}(t)$ est vraie pour tout arbre t vérifiant $(P3)$ et $(P4)$.

6. La famille des arbres bicolores $\mathcal{B}$ est une famille d'arbres qui vérifient en particulier ($P3$) et ($P4$). Ainsi, tout arbre $t \in s\mathcal{B}$ vérifie donc les deux inégalités prouvées ci-dessus :

$$n(t) + 1 \geq 2^{b(t)} \text{ et } h(t) \leq 2b(t)$$

On a donc, $\forall t \in \mathcal{B}$:

$$h(t) \leq 2\log_2(n(t) + 1)$$

Ceci montre que $\mathcal{B}$ constitue une famille d'arbres équilibrés au sens de la définition donnée lors du TP précédent.

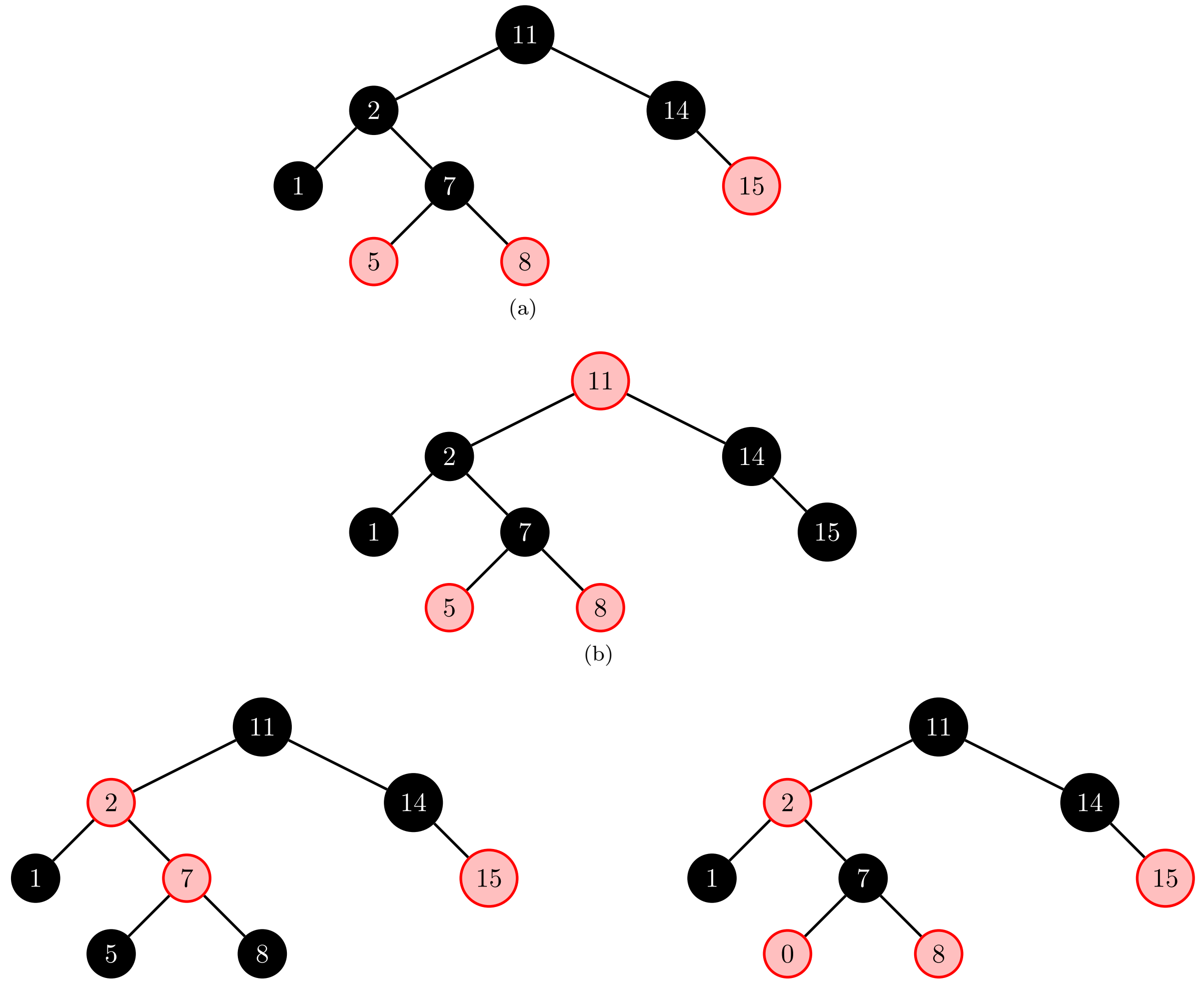
7. Pour les arbres AVL, on a montré que, $\forall t \in \mathcal{A}$:

$$h(t) \leq \underbrace{\frac{\ln(2)}{\ln(\varphi)}}_{\approx 1.44} \log_2(n(t) + 1)$$

Pour les arbres bicolore, on a montré que, $\forall t \in \lfloor$:

$$h(t) \leq 2\log_2(n(t) + 1)$$

La borne d'équilibrage est donc un peu plus fine pour les arbres AVL qui garantissent donc un équilibrage légèrement meilleur que pour les arbres bicolores. De plus, temporellement, les arbres AVL, bien qu'ils nécessitent quelques calculs de facteurs d'équilibrage, ne nécessitent pas de répercuter la correction au delà du premier nœud déséquilibré au dessus de l'insertion. Une implémentation itérative des arbres AVL est particulièrement efficace (cf corrigé TP précédent)
Par contre, les **arbres AVL sont légèrement plus coûteux spatialement** car ils nécessitent le stockage de la hauteur de chaque nœud (un entier de 4 octets) pour le calcul des facteurs d'équilibrage, là où les arbres bicolores ne nécessitent qu'un bit (rouge ou noir) (en fait un octet, en lien avec les contraintes machines) de stockage supplémentaire



(a) (b) (c) (d)

FIGURE 2 – Arbres qui ne sont pas des arbres rouge-noir.

II. Insertion dans un arbre rouge-noir

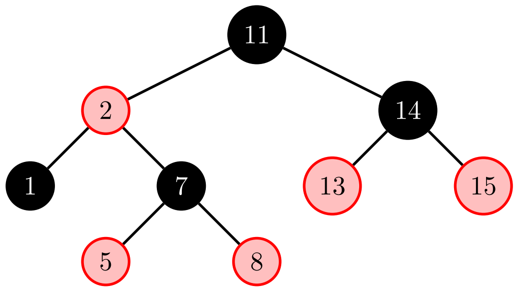
La **recherche** d'un élément dans un arbre rouge-noir se fait exactement comme dans un arbre binaire de recherche. Il suffit d'ignorer la couleur des nœuds. L'**insertion** d'un élément dans un arbre rouge-noir se fait également de la même manière : on recherche la position d'insertion et on y ajoute une feuille contenant l'élément à insérer. **On convient que l'on colorie cette nouvelle feuille en rouge.**

Exercice 2 (Problème rouge-rouge).

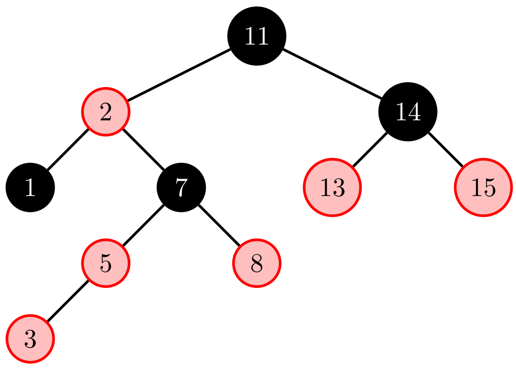
1. Dessiner l'arbre obtenu après ajout de l'élément 13 dans l'arbre de l'exemple 1. L'arbre obtenu est-il toujours un arbre rouge-noir ?
2. Dessiner l'arbre obtenu après ajout de l'élément 3 dans l'arbre obtenu précédemment. Cette arbre est-il encore un arbre rouge-noir ?
3. Quelle(s) propriété(s) des arbres rouge-noir peuvent être violées par l'ajout d'une feuille rouge à la bonne position ? Justifier.

Corrigé de l'exercice 2. *[Retour à l'énoncé]*

1. Voici l'arbre de l'exemple après ajout de 13 : l'arbre est toujours un arbre rouge-noir.



2. Voici l'arbre de l'exemple après ajout de 3 : l'arbre n'est plus un arbre rouge-noir car 5 est rouge et a une fils rouge, donc la propriété $(P3)$ n'est plus vraie.



Les réponses de l'exercice précédent indiquent qu'il n'y a que deux violations possibles des propriétés d'arbre rouge-noir suite à l'insertion d'un élément.

a) La racine est rouge.

b) Un nœud rouge a un fils rouge ; ce nœud problématique est unique

Notons que la racine ne peut être rouge que si l'arbre était initialement vide, cependant ce premier cas pourra également apparaître lors des corrections que nous allons effectuer pour corriger le problème b). Dans le cas b) le nœud qui a un fils rouge est nécessairement le père de la feuille qui vient d'être insérée, mais on va faire **remonter** le problème vers la racine dans le processus de correction, qui pourra donc être déplacé à d'autres endroits par la suite. Le cas b) est dorénavant appelé un problème **rouge-rouge**.

Exercice 3 (Résolution du problème rouge-rouge).
On considère un arbre bicolore valide.
On y insère un nœud rouge : d'après l'exercice précédent, l'arbre obtenu vérifie toutes les propriétés des arbres bicolores sauf éventuellement a) et b) ci-dessus.

1. Justifier que si on est dans le cas a) alors il suffit de colorier la racine en noir pour obtenir un arbre rouge-noir correct.

2. On suppose désormais que l'on est dans le cas b) sans être dans le cas a). Montrer que le père du nœud rouge ayant un fils rouge existe et est noir.

3. Il y a alors quatre configurations possibles selon si les deux nœuds rouges du problème **rouge-rouge** sont fils gauches ou fils droits. La figure ci-après donne les deux configurations correspondant au cas où le nœud rouge ayant un fils rouge est fils gauche de son père noir. Les sous-arbres a, b, d et d sont quelconques mais vérifient les propriétés 1), 3) et 4) des arbres rouge-noir. Dessiner de même les deux configurations (3) et (4) symétriques, correspondant au cas où le nœud rouge ayant un fils rouge est fils droit de son père noir.

4. On se propose de corriger ces quatre cas en modifiant l'arbre comme indiqué sur la figure 3. Une telle transformation sera appelée **correction rouge** par la suite.

   a. Justifier que l'arbre de la figure 3, et donc qu'une correction rouge, peut être obtenue par une rotation gauche, droite, gauche-droite ou droite-gauche à partir de chacune des quatre configurations, suivie d'une simple recoloration des nœuds. Préciser quelle rotation ou double rotation correspond à quel cas et quelle recoloration est effectuée.

   b. Justifier qu'une correction rouge conserve les propriétés $(P1)$ et $(P4)$ d'arbre rouge-noir et corrige le problème b) correspondant à la propriété $(P3)$ pour le sous-arbre obtenu.

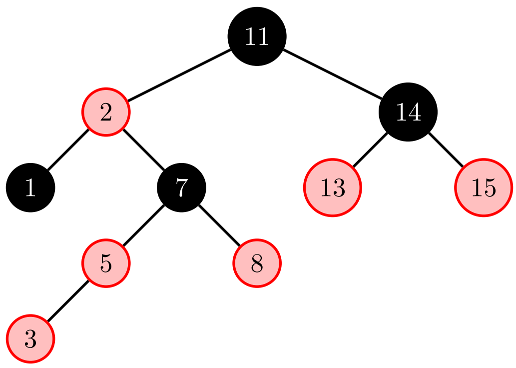
   c. Cependant, on peut ainsi avoir créé un nouveau un problème **rouge-rouge** entre la nouvelle racine rouge y du nouveau sous-arbre et son éventuel père. Il suffit alors de recommencer avec une correction rouge plus haut. Justifier que ce processus termine.

   d. Appliquer cette transformation complète à l'arbre de l'exercice précédent pour obtenir un arbre rouge-noir correct, en n'oubliant pas de recolorer la racine en noir à la fin si besoin pour garantir $(P2)$.

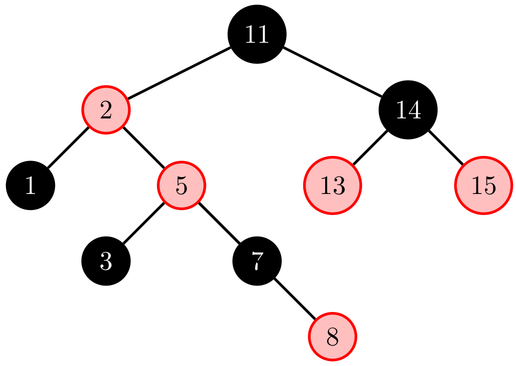
Remarque. Le cas de la suppression est plus délicat car la suppression d'un nœud noir peut changer la hauteur noire $b(t)$ de certains sous-arbres.

Corrigé de l'exercice 3. [Retour à l'énoncé]

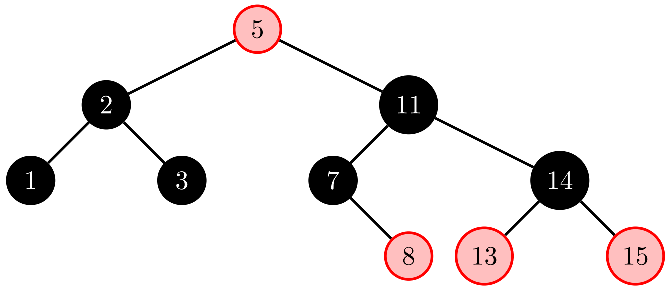
Corrigé de la dernière question :



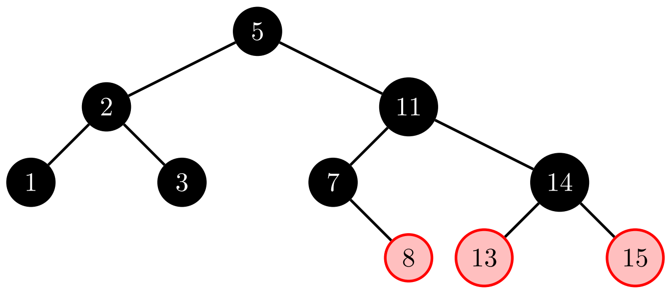
Rotation droite autour de 7 et coloriage adapté, 3 devient noir :



Rotation gauche autour de 2 puis droite autour de 11 puis coloriage adapté, 2 devient noir



Et on n'oublie pas de colorer la racine en noir tout à la fin :



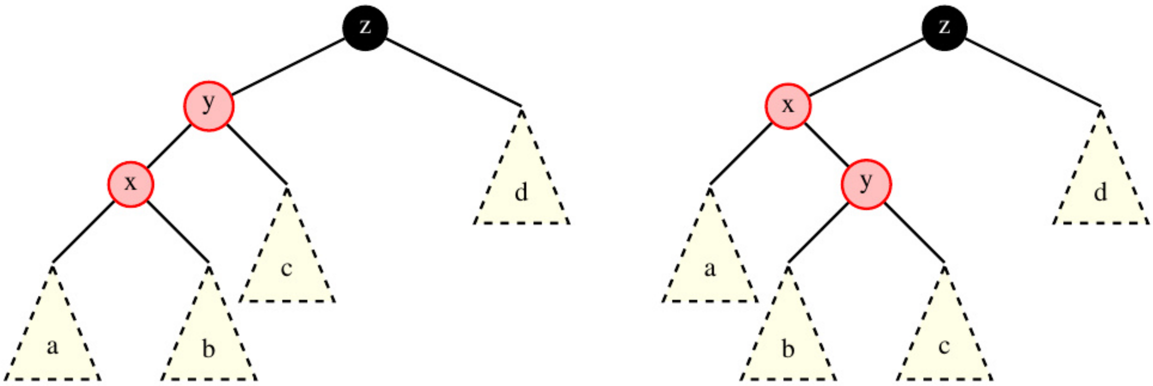


FIGURE 3 – Configurations (1) et (2).

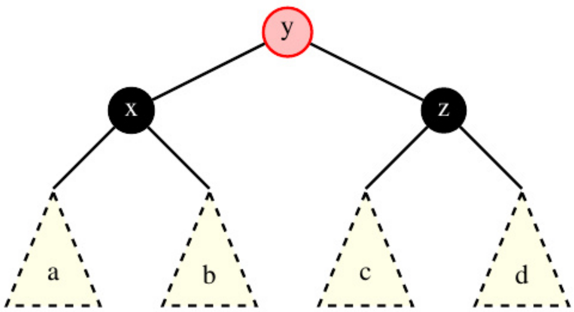


FIGURE 4 – Correction rouge proposée aux configurations (1), (2), (3) et (4).

FIGURE 3 – Configurations possibles de problèmes rouge-rouge (2 sur 4) lors d'une insertion et correction