

Feuille de TD n°3

Un arbre binaire est dit *H-équilibré* si en tout noeud de l'arbre les hauteurs des sous-arbres gauche et droit diffèrent au plus de 1. Si, de plus, c'est un arbre binaire de recherche, on l'appelle un *arbre AVL*.

Exercice 1 (Arbres de Fibonacci). On définit la suite d'*arbres de Fibonacci* par récurrence de la manière suivante :

- $A_0 = \emptyset$ (arbre vide),
- A_1 est réduit à un seul noeud,
- pour $n \geq 2$, A_n est l'arbre binaire dont le sous-arbre gauche est A_{n-1} et le sous-arbre droit est A_{n-2} .

1. Construire A_2, A_3, A_4, A_5 .
2. Montrer que les arbres de Fibonacci sont H-équilibrés.
3. Exprimer la hauteur h_n et le nombre de noeuds N_n de A_n à l'aide des nombres de la suite de Fibonacci F_n . On rappelle que cette suite est définie par récurrence par $F_0 = 0, F_1 = 1$ et pour $n \geq 2, F_n = F_{n-1} + F_{n-2}$.

Exercice 2 (Suppression dans un arbre AVL). La suppression d'un élément dans un arbre AVL s'effectue comme dans le cas d'un arbre binaire de recherche classique. Cependant, l'arbre résultant n'est pas forcément H-équilibré. Il faut donc rajouter une étape de rééquilibrage de l'arbre après suppression. Il s'effectue à l'aide des quatre rotations gauche, droite, gauche-droite et droite-gauche.

```
RG(arbre A) {
  arbre Aux
  Aux <- Sous_Arbre_Droit(A)
  Sous_Arbre_Droit(A) <- Sous_Arbre_Gauche(Aux)
  Sous_Arbre_Gauche(Aux) <- A
  A <- Aux
}
```

```
RD(arbre A) {
  arbre Aux
  Aux <- Sous_Arbre_Gauche(A)
  Sous_Arbre_Gauche(A) <- Sous_Arbre_Droit(Aux)
  Sous_Arbre_Droit(Aux) <- A
  A <- Aux
}
```

```
RGD(arbre A) {
  RG(Sous_Arbre_Gauche(A))
  RD(A)
}
```

```
RDG(arbre A) {
  RD(Sous_Arbre_Droit(A))
  RG(A)
}
```

1. Essayer ces rotations sur de petits exemples.
2. Quelles sont les conditions pour pouvoir appliquer ces rotations ?
3. Expliciter les changements apportés par ces rotations sur les hauteurs des sous-arbres.
4. En déduire une méthode de rééquilibrage d'un arbre AVL (après suppression d'un élément).

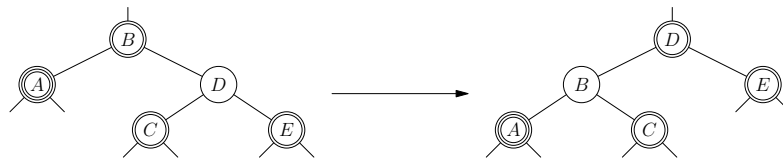
On s'intéresse à l'algorithme de suppression dans un arbre rouge-noir. Si un nœud n'a qu'un fils ou s'il n'en a pas du tout, on complète sa filiation en lui octroyant un ou deux fils vides, que l'on considère comme étant noirs (ce qui préserve les propriétés d'arbre rouge-noir.) Pour supprimer un nœud n non vide, on commence par le supprimer comme dans le cas d'un arbre binaire de recherche, avec les couleurs explicitées ci-dessous.

- Si n n'a que des fils vides, on le remplace simplement par un nœud vide (et on supprime ses fils vides).
- Si n a un seul fils non vide, on le remplace par ce fils.
- Si n a deux fils non vides, on le remplace par le plus petit nœud de son sous-arbre droit (c'est-à-dire l'élément immédiatement supérieur en valeur), que l'on supprime selon les règles ci-dessus. Le nouveau nœud garde la couleur qu'avait n .

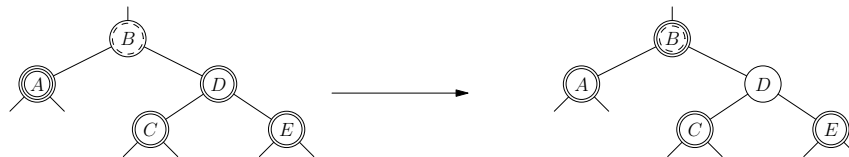
Dans le dernier cas, comme on ne modifie pas la couleur du nœud remplacé, on se ramène à l'un des deux premiers cas. Dans ces deux premiers cas, les propriétés d'arbre rouge-noir peuvent se retrouver violées : plus précisément, si le nœud supprimé était rouge, on a toujours un arbre rouge-noir. Mais s'il était noir, les chemins qui passaient par ce nœud se retrouvent avec un noir de moins. Pour corriger cela, on autorise l'arbre à avoir un nœud « doublement noir » : celui-ci compte comme deux nœuds noirs. Puis on réalise des opérations visant à supprimer ce nœud doublement noir.

Si on a supprimé le nœud n (qui avait un ou aucun fils non vide) et que celui-ci était noir, on considère comme doublement noir le fils qui a pris sa place, s'il n'avait qu'un fils non vide, et le fils vide par lequel on l'a remplacé, s'il avait deux fils vide. Puis on réalise des opérations selon les quatre cas suivants.

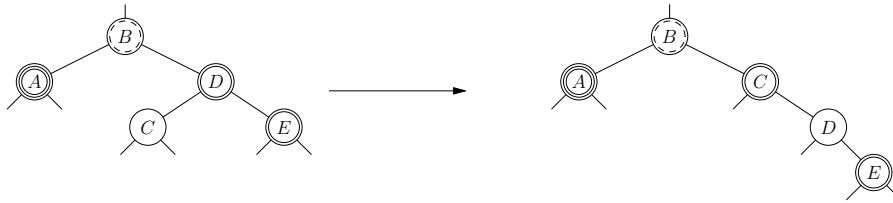
- Cas 1. Le frère du nœud doublement noir est rouge. On se ramène alors au cas où ce frère est noir en faisant la rotation et les changements de couleur suivants (les nœuds noirs sont représentés avec un double cercle, le nœud doublement noir avec un triple cercle, les nœuds rouges avec un cercle simple ; dans les schémas suivants, les nœuds de couleur indéterminée sont représentés avec un cercle pointillé en plus du ou des cercles pleins).



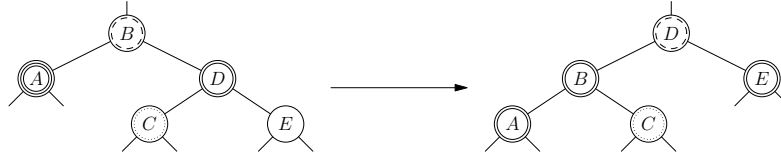
- Cas 2. Le frère et les neveux sont noirs ou n'existent pas. On colorie alors le frère en rouge, et, si le père est rouge, on le colorie en noir, sinon, on lui attribue l'étiquette « doublement noir » et on recommence.



- Cas 3. Le frère est noir, le neveu droit est noir ou n'existe pas, le neveu gauche est rouge. On réalise alors une rotation et des changements de couleur comme ci-dessous, ce qui nous amène dans la situation du cas 4.



- Cas 4. Le frère est noir et le neveu droit est rouge. On réalise alors la rotation et les changements de couleur suivants, ce qui nous débarrasse de l'étiquette « doublement noir ».



La boucle se termine soit quand on arrive dans le cas 4, soit quand la racine de l'arbre est doublement noire, auquel cas, en la laissant simplement noire, on a bien un arbre rouge-noir.

Exercice 3. Supprimer 1 de l'arbre rouge-noir suivant, en détaillant les opérations effectuées.

