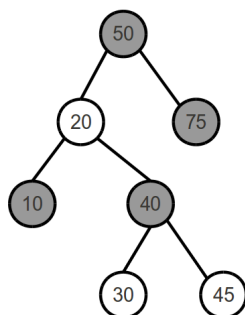


Feuille d'exercices no. 6

Arbres rouge et noir

Exercice 1. Dessiner l'arbre binaire de recherche complet de hauteur 3 contenant les valeurs $\{1, 2, \dots, 15\}$. Ajouter les feuilles NIL et colorier ensuite les noeuds de trois manières différentes pour que les hauteurs noires des arbres rouge et noir résultants soient respectivement 2, 3 et 4.

Exercice 2. On considère l'arbre rouge et noir suivant (les noeuds remplis correspondent aux noeuds noirs).



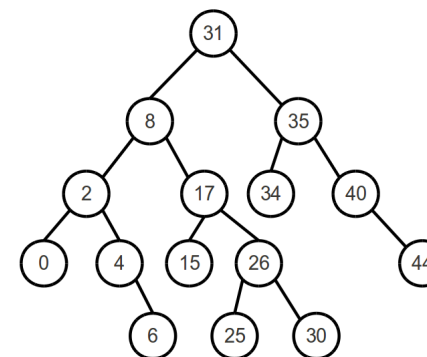
1. Vérifier que cet arbre respecte bien les propriétés définissant les arbres rouge et noir.
2. Insérer 35 dans l'arbre ci-dessus. On détaillera les opérations effectuées pour équilibrer l'arbre.

Exercice 3. Construire l'arbre rouge et noir obtenu par des insertions successives des entiers 1, 12, 7, 15, 18, 9, 20, 8, 13. Ensuite, supprimer la racine de l'arbre, le sous-arbre gauche de l'arbre obtenu, et le sous-arbre droit de l'arbre obtenu par la dernière opération.

Exercice 4. Ecrire un algorithme pour calculer la hauteur noire de chaque noeud d'un arbre rouge et noir.

Exercice 5. Pour montrer que tout arbre AVL A peut être colorié pour obtenir un arbre rouge et noir, nous considérons l'algorithme suivant. D'abord, on colorie tous les noeuds de A en noir. Ensuite, pour tout noeud X , si $h(X)$ est impair alors on colorie $X.G$ (resp. $X.D$) en rouge si $X.G \neq NIL$ et $h(X.G)$ est pair (resp. $X.D \neq NIL$ et $h(X.D)$ est pair), où $h(X)$ est la hauteur du sous-arbre de racine X .

1. Vérifier que l'algorithme indiqué est correct en l'appliquant sur l'AVL ci-dessous :



2. Écrire une procédure $ColorAVL(AVL A)$ qui colorie un AVL A pour obtenir un arbre rouge et noir.

Pour aller plus loin

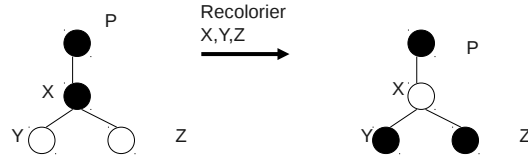
Exercice 6. Dans un ARN, les opérations d'insertion ou suppression d'un élément e donnent lieu à une descente dans l'arbre pour trouver l'endroit où e doit être inséré, respectivement pour trouver l'endroit où e apparaît dans l'arbre. Lors de l'insertion, un deuxième parcours de la branche entre la racine et cet endroit peut être nécessaire, pour rééquilibrer et recolorier l'arbre. Une idée alternative, évitant le double parcours de la branche, est de profiter de la descente dans l'arbre pour faire des changements assurant que, lors de l'insertion/suppression de l'élément e il n'y aura pas besoin de remonter dans l'arbre.

Nous nous concentrons dans cet exercice sur l'insertion d'un élément e . Lors de la descente dans l'arbre, soit X le sommet courant (celui dont la valeur est comparée à e pour savoir s'il faut continuer la recherche de e dans le sous-arbre gauche ou droit de X). En fonction de la configuration autour de X dans l'ARN, trois cas gênants peuvent apparaître, qui sont traités comme indiqué dans la Figure ci-dessous.

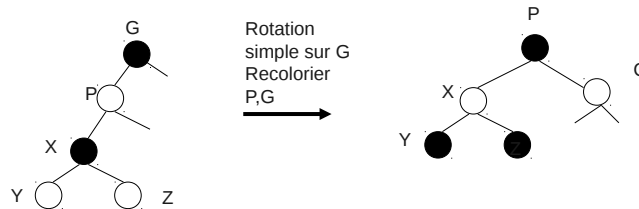
Nous remarquons que les cas gênants sont ceux où X est noir et ses deux fils Y, Z sont rouges, puisque si e était inséré comme fils de X ou de Y , son oncle serait rouge et on serait donc dans le cas 2 de l'insertion (celui qui fait remonter le problème vers le haut).

Insertion « Top-down » dans un ARN

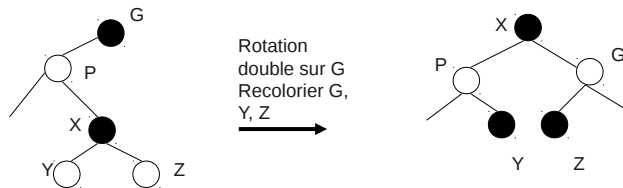
Cas 1
P est noir



Cas 2
P est rouge
X & P sont tous les deux des fils gauches (similaire, des fils droits)



Cas 3
P est rouge
X et P sont des fils de types opposés (l'un est fils gauche, l'autre fils droit)



On demande :

1. Reprendre la partie “insertion” de l’exercice 3, en utilisant l’insertion top-down.
2. Discuter le besoin d’avoir, dans chaque sommet de l’arbre, un pointeur sur le père de ce sommet, dans les deux cas d’insertion (bptm-up et top-down).