

# IFT2015 Hiver 2009 — Devoir 3

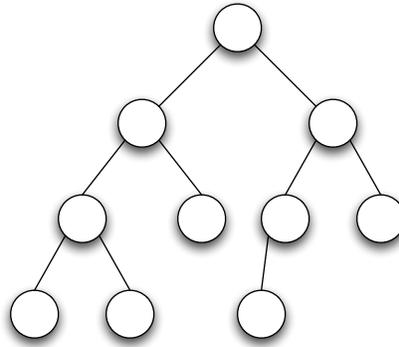
Miklós Csűrös

18 mars 2009

À remettre au cours du 30 mars. Un rapport soumis par courriel (à csuros@iro...) en un seul fichier PDF vaut  $(1 + \sqrt{5})$  points de boni.

## 1 Arbre AVL (30 points)

Un arbre AVL contient les clés 1,2,3,4,5,6,7,8,9,10. On connaît la structure de l'arbre<sup>1</sup> :



**Clés (10 points).** Remplissez le dessin avec les clés des nœuds.

**Parcours (20 points).** Montrez l'ordre d'énumération de nœuds lors de parcours préfixe, infixe, postfixe et par ordre de niveau (*breadth-first*).

- Dans quel(s) ordre(s) d'énumération peut on calculer la profondeur (niveau) de chaque nœud ?
- Dans quel(s) ordre(s) d'énumération peut on calculer la hauteur de chaque nœud ?
- Montrez la profondeur et la hauteur de chaque nœud dans l'arbre (par la convention chez les arbres AVL, la hauteur d'une feuille null est  $-1$ , et le niveau de la racine est 0).
- Dans quel(s) ordre(s) peut on insérer les clés dans un arbre initialement vide pour obtenir la structure montrée ? **Indice** : on veut éviter les rotations qui sont nécessaires lorsque la propriété AVL est violée.

---

<sup>1</sup>Les feuilles null ne sont pas montrées.

## 2 Arbres rouge, rouge et noir (30+5 points)

Rudolf Bayer a inventé une structure de données sous le nom «symmetric binary B tree» qui est équivalente à l'arbre rouge et noir. Il mentionne une généralisation qu'on considère ici. Un *arbre rouge-rouge-noir* (RRN) est un arbre binaire de recherche avec des feuilles null comme les arbres rouges et noirs, dans lequel les nœuds sont équipés d'un *rrang*. (Ne pas confondre avec le *rang* des arbres RN.) Les *rrangs* des nœuds satisfont les propriétés suivantes.

1. Pour chaque nœud  $x$  excepté la racine,

$$\text{rrang}(x) \leq \text{rrang}(\text{parent}(x)) \leq \text{rrang}(x) + 1.$$

2. Pour chaque nœud  $x$  avec un arrière-grand-parent  $y = \text{parent}(\text{parent}(\text{parent}(x)))$ ,

$$\text{rrang}(x) < \text{rrang}(y).$$

3. Pour chaque feuille  $x$  on a  $\text{rrang}(x) = 0$  et  $\text{rrang}(\text{parent}(x)) = 1$ .

On colorie les nœuds soit rouge soit noir : le nœud  $x$  est rouge si et seulement si il n'est pas la racine et  $\text{rrang}(x) = \text{rrang}(\text{parent}(x))$ . Tous les autres nœuds sont noirs.

**Coloriage et *rrang* (15 points)** Complétez l'énoncé du théorème suivant et démontrez qu'il est vrai en utilisant les propriétés de *rrang*.

**Théorème 1** *Le coloriage d'un arbre rouge-rouge-noir est tel que*

- (i) chaque feuille est colorié par .
- (ii) si le parent d'un nœud  est , alors son grand-parent (s'il existe) est .
- (iii) chaque chemin reliant un nœud à une feuille dans son sous-arbre contient le même nombre de nœuds .

**La hauteur d'un arbre rouge-rouge-noir (15 points)** On veut démontrer que la hauteur d'un arbre rouge-rouge-noir avec  $n$  nœuds internes est  $O(\log n)$ . Il n'est pas trop difficile de démontrer le théorème suivant.

**Théorème 2** *Pour tout nœud  $x$  d'un arbre rouge-rouge-noir, le nombre de nœuds internes dans le sous-arbre enraciné à  $x$  est au moins  $2^{\text{rrang}(x)} - 1$ .*

Vous pouvez utiliser ce théorème sans preuve pour démontrer **b** ici. Pour 5 points de boni, donnez une preuve formelle au théorème 2.

**a (7 points)** Démontrez que

$$\text{hauteur}(x) \leq 3 \cdot \text{rrang}(x) \quad (*)$$

pour chaque nœud  $x$ . (Vous avez besoin du théorème 1 dans la preuve.)

**b (8 points)** Utilisez le théorème 2 et l'équation (\*) pour démontrer que la hauteur d'un arbre RRN avec  $n$  nœuds internes est borné par  $c \lceil \lg(n+1) \rceil$  — remplacez  $c$  par un nombre aussi petit que possible. Comparez ce résultat avec la borne sur la hauteur des arbres RN.