

TD6 : arbres

Exercice 1 : Construction d'un tas

Q 1.1 Construire le tas associé aux valeurs : 5,13,2,25,7,17,20,8,4.

Q 1.2 Identifier un pire des cas pour la construction d'un tas. Quel est le nombre d'échanges réalisé ?

Q 1.3 Identifier un meilleur des cas pour la construction d'un tas. Quel est le nombre d'échanges réalisé ?

Exercice 2 : Construction d'AVL

Construire l'arbre AVL pour la série de valeurs suivante : 5,13,2,25,7,17,20,8,4.

Exercice 3 : Files de priorité

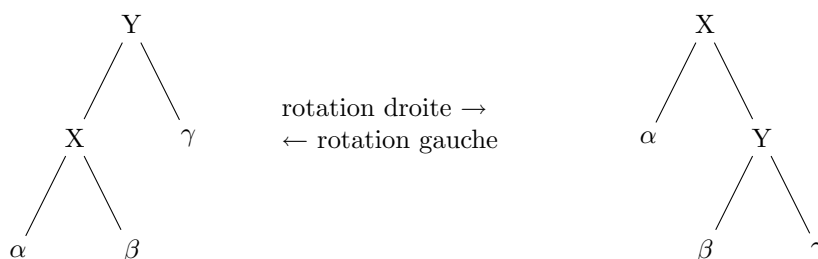
Une file de priorité est une structure de données permettant de gérer un ensemble de données où chaque donnée se voit associée une priorité. Ainsi l'extraction des données se fait par ordre de priorité décroissant. On se sert de ce type de structure de données pour, par exemple, gérer l'ordre d'exécution des tâches dans un système d'exploitation.

Q 3.1 En s'inspirant des structures de données déjà rencontrées, proposer une structure de données permettant d'implanter une file de priorité.

Q 3.2 Proposer une implantation de la fonctionnalité d'extraction de la donnée de priorité maximale.

Q 3.3 Discuter de la complexité de cette fonctionnalité.

Exercice 4 : Calcul de l'équilibre des nœuds dans les AVL



On note $d(n)$ le déséquilibre du nœud n et $d'(n)$ le déséquilibre du nœud n après une rotation. Démontrer les propriétés suivantes :

- Après une rotation droite autour du sommet Y , on a :
 - $d'(X) = d(X) - 1 + \min(d'(Y), 0)$
 - $d'(Y) = d(Y) - 1 - \max(d(X), 0)$
- Après une rotation gauche autour du sommet X , on a :
 - $d'(X) = d(X) + 1 - \min(d(Y), 0)$
 - $d'(Y) = d(Y) + 1 + \max(d'(X), 0)$

Exercice 5 : Expression des rotations

Nous avons vu dans le cours le code pour exécuter des rotations gauche et droite quand les nœuds étaient déclarés ainsi :

```
1 type
2   ELEMENT = CARDINAL;
3   AVL = ^NOEUD;
4   NOEUD = record
5     valeur : ELEMENT;
6     filsg : AVL;
7     filsd : AVL;
8     hauteur : INTEGER;
9   end {NOEUD};
```

On suppose maintenant qu'au lieu de la hauteur du sous-arbre, c'est la valeur du déséquilibre du nœud qui sera stockée :

```
1 type
2   ELEMENT = CARDINAL;
3   AVL = ^NOEUD;
4   NOEUD = record
5     valeur : ELEMENT;
6     filsg : AVL;
7     filsd : AVL;
8     disequilibre : INTEGER;
9   end {NOEUD};
```

Q 5.1 Ecrire le code de l'opération de rotation droite avec ce nouveau type pour les nœuds.

Q 5.2 Combien de bits d sont nécessaires pour stocker la valeur du déséquilibre ?

On suppose maintenant que les entiers sont stockés sur w bits et que les valeurs associées aux nœuds (toujours des entiers) sont telles qu'elles nécessitent au plus $w - d$ bits pour être stockés.

Q 5.3 Proposer un stockage intelligent de la valeur du nœud et du déséquilibre. Donner la nouvelle déclaration du type NOEUD.

Q 5.4 Ecrire le code de la fonction :

function valeur (a : AVL) : ELEMENT;

Q 5.5 Ecrire le code de la fonction :

function disequilibre (a : AVL) : INTEGER;