Devoir Domestique d'Algorithmique Avancée

Travail à effectuer en binôme et à remettre le lundi 11 décembre 2006, sous forme d'un document imprimé

Un arbre 2-3 est un *arbre de recherche*. Cela signifie qu'il sert à stocker de l'information, et que celle-ci est constituée d'éléments d'un ensemble totalement ordonné. De plus il permet de réaliser sur cette information les opérations *Recherche*, *Ajout*, *Suppression*.
Plus précisément, il est tel que
- chaque noeud interne a exactement 2 ou 3 fils, et constitue ainsi un 2-noeud ou un 3-noeud ;
- toutes les feuilles sont à même distance de la racine.

Chaque feuille de l'arbre contient un élément d'information de l'ensemble à représenter par l'arbre 2-3, et les éléments sont rangés dans les feuilles de gauche à droite par ordre croissant. Chaque noeud interne contient, pour faciliter la recherche, 2 valeurs : le plus grand élément du premier sous-arbre de ce noeud, et le plus grand élément du second sous-arbre ; on notera $<g,m>$ ce couple de valeurs. La figure 1 représente un arbre 2-3 de hauteur 3 dont les informations sont, pour simplifier, prises dans un sous-ensemble d'entiers naturels.

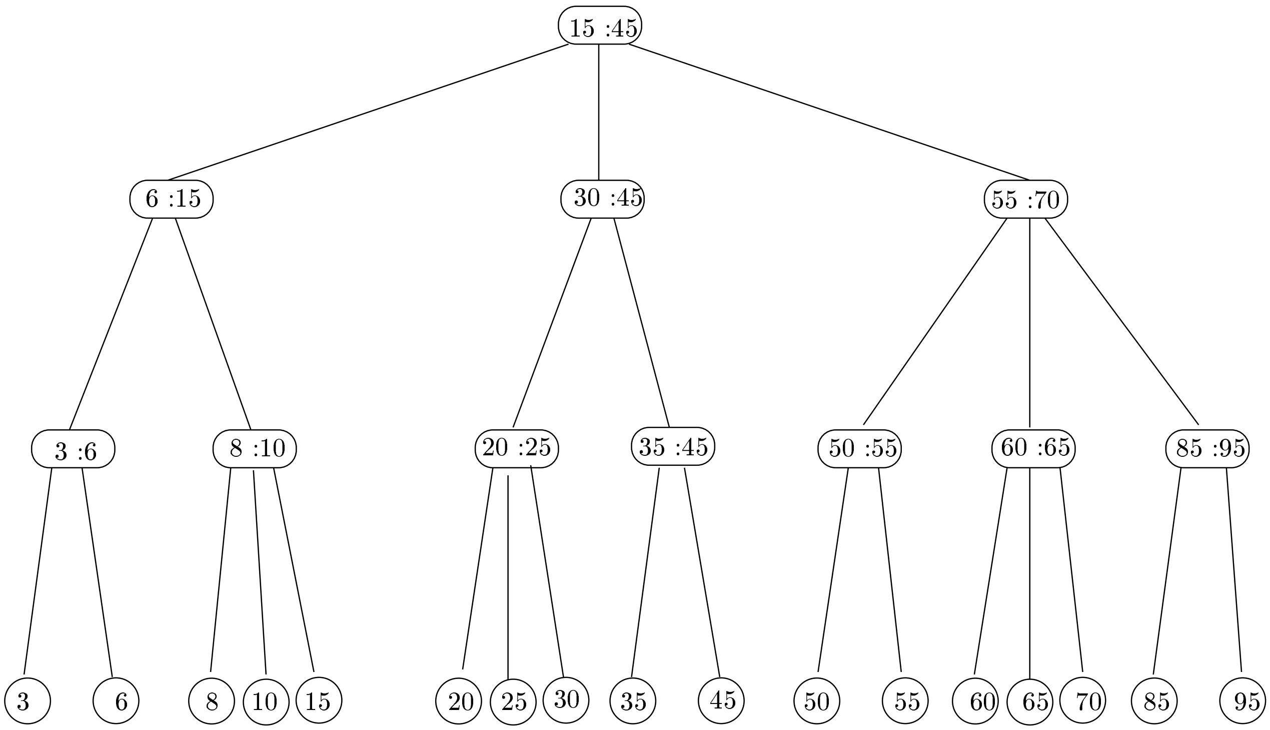

FIG. 1 – Arbre 2-3 de hauteur 3.

Le travail demandé est l'étude de la représentation des arbres 2-3 et des algorithmes de Recherche, d'Ajout et de Suppression d'un élément d'information. On précisera pour chaque algorithme le principe de la méthode employée, l'expression des structures de données et de l'algorithme en EXALGO, ainsi que l'analyse de la complexité, en précisant pour le calcul de cette dernière le paramètre choisi pour mesurer la taille de la donnée, ainsi que les opérations dont on compte les exécutions.

Préliminaires. (4 points)

Question 1 : dessiner toutes les formes (sans indiquer les valeurs) des arbres 2-3 de hauteur 2.
Question 2 : proposer deux exemples d'arbre 2-3 ayant même nombre de feuilles et des hauteurs différentes.
Question 3 : donner des relations entre la hauteur d'un arbre 2-3 et le nombre d'informations stockées par cet arbre (nombre de feuilles).

Représentation EXALGO de l'arbre 2-3. (6 points)

En supposant que l'information stockée est une valeur de type entier, donner la définition EXALGO du type arbre 2-3, qui soit compatible avec les opérations statiques (Recherche) aussi bien qu'avec les opérations dynamiques qui modifient la taille de l'arbre (Ajout, Suppression).

Recherche. (4 points)

Ecrire la primitive Recherche, qui retourne dans le cas général, si la valeur cherchée est trouvée, un booléen à vrai, et la feuille concernée, et si la valeur est absente, un booléen à faux.
On rappelle que l'on fait l'hypothèse que toutes les informations stockées sont distinctes.

Ajout (6 points)

On demande une méthode d'ajout qui conserve à l'arbre concerné la qualité d'arbre 2-3. On suppose que l'arbre 2-3 possède au moins un sommet.
Méthode proposée pour l'ajout d'une feuille x contenant la nouvelle valeur :
Tout d'abord on vérifie l'absence de la valeur dans l'arbre, puis on procède à l'ajout.
cas singulier : l'arbre a un seul sommet. Soit s ce sommet qui est à la fois racine et feuille. On crée une nouvelle racine r, et une nouvelle feuille x, contenant la valeur à insérer, et on place s comme feuille gauche de r et x comme feuille droite si la nouvelle valeur ajoutée est supérieure à celle contenue dans le sommet initial s, sinon on intervertit l'ordre des fils de la racine r. Bien entendu on doit calculer le couple $<g : m>$ pour la racine r.
cas général : soit p le sommet interne destiné à être père de la nouvelle feuille.

- si p n'a que 2 fils , les feuilles y_1, et y_2, on ajoute la nouvelle feuille x en l'insérant à sa place selon l'ordre des valeurs, parmi y_1, et y_2. On met à jour éventuellement $<g : m>$ pour p (et peut-être pour d'autres sommets).
- si p a déjà 3 fils, on ajoute provisoirement à p, une nouvelle feuille à la bonne place, et p devient provisoirement un 4-noeud. On partage alors p en deux 2-noeuds, p et p', puis

p' est à son tour inséré dans le père de p. Le processus de réajustement peut remonter jusqu'à la racine (ne pas oublier de mettre à jour les couples $<g, m>$ concernés).
Question 1 : Illustrer les différents cas possibles d'ajouts par des exemples.
Question 2 : Ecrire en EXALGO la procédure d'ajout. Calculer sa complexité maximale.

Suppression (Bonus 4 points)

On suggère la méthode générale suivante :
Si le père de la feuille à supprimer est un 3-noeud, il devient un 2-noeud et il suffit de mettre à jour éventuellement les valeurs des couples $<g, m>$ sur les ascendants.
Si le père de la feuille à supprimer est déjà un 2-noeud, 2 solutions sont possibles : si ce père a un frère en excédent (3-noeud) récupérer un fils de ce frère, sinon fusionner avec un frère ayant seulement 2 fils (cette fusion enlève de nouveau un fils à son père ...).
Question 1. Développer la méthode en précisant les différents cas ou options à choisir, et en les illustrant par des exemples.
Question 2. En déduire une procédure de suppression en EXALGO.
Question 3. Préciser sa complexité.