TD numéro 2
Complexité

Programmation impérative avancée, ENSIIE

Semestre 2, 2015–16

Exercice 1 : Bases sur les arbres

1. On considère qu'un arbre réduit à un nœud est de hauteur 1. Combien de noeuds peut contenir un arbre de hauteur h ?
2. Écrire les fonctions `hauteur` et `nombre_de_noeuds` et donner leur complexité.
3. On rappelle qu'un arbre t est dit bien équilibré si la différence de hauteur entre ses sous-arbres est au plus de 1, et si ses sous-arbres sont bien équilibrés. En supposant qu'on ne stocke pas la hauteur de l'arbre dans les nœuds, donner la fonction `est_bien_equilibre` ainsi que sa complexité.

Exercice 2 : Arbres de Fibonacci

1. Définir par récurrence une famille d'arbres, indexée par leur hauteur, qui contiennent le moins de nœuds possible tout en étant bien équilibrés. Dessiner ces arbres jusqu'à la hauteur 5.
2. En déduire une relation de récurrence sur le nombre minimal de nœuds n_h dans un arbre bien équilibré de hauteur h.
3. On rappelle la suite de Fibonacci :

$$\begin{aligned} f_0 &= 0 \\ f_1 &= 1 \\ f_i &= f_{i-1} + f_{i-2} \qquad\qquad i \geq 2 \end{aligned}$$

   Exprimer n_h en fonction de cette suite.
4. On admettra que les termes de la suite de Fibonacci peuvent s'écrire $f_i = \frac{1}{\sqrt{5}}(\Phi^i - \Phi^{-i})$ avec $\Phi = \frac{1+\sqrt{5}}{2}$. En déduire une borne minimale pour le nombre de nœud d'un arbre bien équilibré de hauteur h.
5. Encadrer la hauteur d'un arbre bien équilibré en fonction de son nombre de nœuds.
6. En déduire la complexité dans le pire des cas des opération de recherche, d'insertion et de suppression dans un arbre AVL.