

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Sections d'Informatique et de Systèmes de Communication

Série d'exercices 5

19 Octobre 2007

1. Arbres AVL

Ajouter les éléments suivants (dans l'ordre donné) dans un arbre AVL (initialement vide) de telle façon qu'il reste AVL :

12, 9, 123, 3, 0, 23, 98, 2, 11, 7

Dessiner l'arbre après chaque insertion.

2. Hashing

Nous aimerions stocker des noms dans un tableau de hachage de taille 26. Pour le faire nous identifions les lettres de l'alphabet avec des nombres : A = 0, B = 1, ..., Z = 25. Nous avons le choix entre deux fonctions de hachage :

$$\begin{aligned}
 h_1 : \{A, \dots, Z\}^+ &\rightarrow \{0, \dots, 25\} \\
 (a_1, \dots, a_n) &\mapsto a_1 \\
 \text{et } h_2 : \{A, \dots, Z\}^+ &\rightarrow \{0, \dots, 25\} \\
 (a_1, \dots, a_n) &\mapsto a_1 + a_3 \pmod{26}.
 \end{aligned}$$

(Nous supposons que les noms sont assez longs pour que h_2 soit défini.)

Voici la liste des noms à traiter avec les valeurs des fonctions h_1 et h_2 correspondantes :

Nom	h_1	h_2	Nom	h_1	h_2
Alfo	0	5	Jean-Marie	9	9
Amel	0	4	Joachim	9	9
Amin	0	8	John	9	16
Anthony	0	19	Jose	9	1
Antoine	0	19	Klaus-Dieter	10	10
Bernard	1	18	Manuel	12	25
Boris	1	18	Marco	12	3
Charles	2	2	Michel	12	14
Denis	3	16	Otto	14	7
Diego	3	7	Peter	15	8
Dominique	3	15	Philippe	15	23
Erik	4	12	Robert	17	18
Eva	4	4	Sacha	18	20
Gerard	6	23	Stephan	18	22
Hichem	7	9	Sylvain	18	3
Ian	8	21	Thomas	19	7
Jacques	9	11	Tudor	19	22
Jean-François	9	9			

a) Écrire les tables de hachage correspondant à l'utilisation de h_1 et h_2 respectivement.

- b) Est-ce que l'une des deux tables de hachage est meilleure que l'autre ? En quel sens ?
- c) Est-ce que la différence est due uniquement au hasard, ou voyez vous une raison pour laquelle une des fonctions de hachage pourrait être préférable à l'autre pour stocker des prénoms ?

3. Algorithmes récursifs : Le diamètre d'un arbre binaire

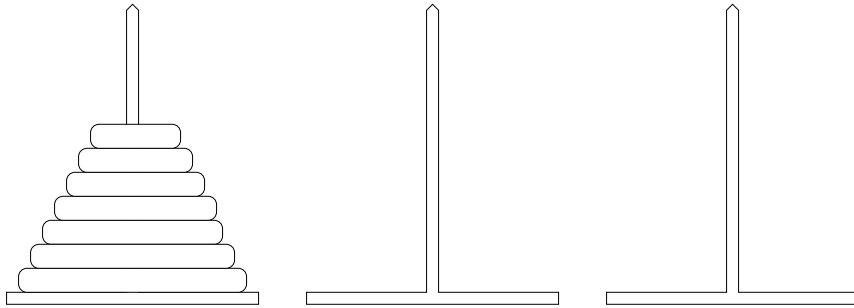
Soit $G = (V, E)$ un graphe. Pour deux sommets $a, b \in V$, on définit la *distance* $d(a, b)$ comme étant la longueur du plus court chemin de a à b . Le *diamètre* de G est défini comme étant la distance maximale dans le graphe, i.e.

$$\text{diam}(G) := \max_{a, b \in V} d(a, b).$$

- a) Soit T un arbre binaire, et soient T_1 et T_2 ses sous-arbres gauche et droit. Exprimer le diamètre puis la hauteur de T en fonction des diamètres et hauteurs de T_1 et T_2 .
- b) Construire un algorithme qui calcule le diamètre d'un *arbre binaire* en temps linéaire (en le nombre de sommets).
(Indice : Il faut construire par récurrence un algorithme qui, étant donné un arbre binaire, retourne sa hauteur et son diamètre en temps linéaire).

4. Les tours de Hanoi

On considère le problème suivant : On a n disques de différents diamètres qui se trouvent sur un bâton. Les disques sont ordonnés selon leur diamètre, avec le plus grand disque au fond. On dispose aussi de deux bâtons, initialement vides.



On aimerait déplacer la tour du premier bâton vers le dernier bâton. Le seul mouvement admis est d'enlever le plus haut élément d'une tour et de le rajouter sur un bâton vide ou un bâton dont le plus haut disque a un diamètre plus grand. Un disque ne peut donc jamais être posé sur un disque de plus petit diamètre.

- a) Trouvez une procédure récursive qui donne la suite de mouvements à effectuer pour déplacer une tour de taille n .
- b) Calculer le nombre de mouvements qui sont effectués pour déplacer une tour de taille n . (Indication : Donner d'abord une formulation récursive, ensuite prouver la formule explicite par induction.)
- c) Montrer que cet algorithme est optimal, i.e. qu'il n'en existe pas un meilleur. (Indication : Considérer l'étape de déplacement du plus grand disque. Qu'est-ce qui est arrivé avant et qu'est-ce qui reste nécessairement à faire ?)