

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

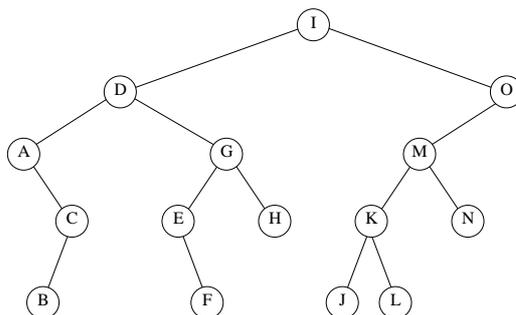
Sections d'Informatique et de Systèmes de Communication

Série d'exercices 4

18 Octobre 2010

1. Arbres

- a) Donner les suites de sommets obtenues en parcourant l'arbre binaire suivant en l'ordre preorder, inorder et postorder.



- b) Vérifier qu'il s'agit bien d'un arbre de recherche binaire pour l'ordre alphabétique.
- c) Prouver qu'un arbre binaire est un arbre de recherche si et seulement si la suite des valeurs obtenue en prenant les clés dans un parcours inorder est croissante.
- d) Effacer le sommet D de manière à préserver la structure d'arbre de recherche.
- e) Écrire un algorithme pour effacer dans un arbre de recherche. Donner un argument qui justifie cet algorithme.
- f) Soit T un arbre binaire de hauteur h . Montrer que T peut avoir au plus $2^{h+1} - 1$ sommets.

2. Graphes

- a) Si la première ligne de la matrice d'adjacence d'un graphe non orienté G ne contient que des 1 alors G est connexe. Vrai ou Faux? (Justifier.)
- b) Dessiner un arbre binaire dans lequel tous les sommets ont degré 0 ou 2, mais qui n'est pas un arbre AVL.
- c) Soit T un arbre binaire avec n sommets. Quelle est la plus petite hauteur que peut avoir T ? Quel est la plus grande hauteur que peut avoir T ?
- d) Soit G un graphe non orienté avec n sommets, et dans lequel tous les sommets ont degré d (un tel graphe est appelé d -régulier). Nous supposons aussi que la diagonale de sa matrice d'adjacence ne contient que des 0.

Montrer que si n est impair alors d doit être pair. (*Indice:* compter le nombre d'arêtes.)

3. 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.

4. 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
Alfio	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			

- Écrire les tables de hachage correspondant à l'utilisation de h_1 et h_2 respectivement.
- Est-ce que l'une des deux tables de hachage est meilleure que l'autre? En quel sens?
- 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?