

# ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Sections d'Informatique et de Systèmes de Communication

Série d'exercices 5

19 Octobre 2009

## 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:

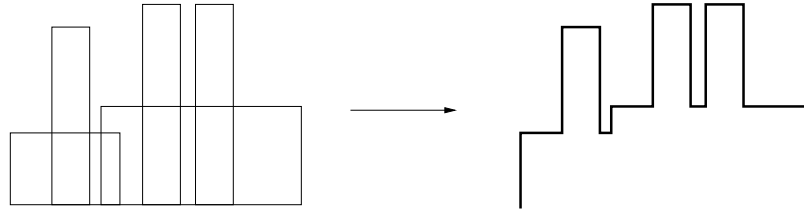
Nom	$h_1$	$h_2$	Nom	$h_1$	$h_2$
Alfio	0	5	Jean-Marie	9	9
Amel	0	4	Joachim	9	9
Amin	0	8	John	9	16
Anthony	0	19	Jose	9	1
Antoine	0	19	Klaus-Dieter	10	10
Bernard	1	18	Manuel	12	25
Boris	1	18	Marco	12	3
Charles	2	2	Michel	12	14
Denis	3	16	Otto	14	7
Diego	3	7	Peter	15	8
Dominique	3	15	Philippe	15	23
Erik	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			

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. Diviser pour régner: Le problème du skyline**

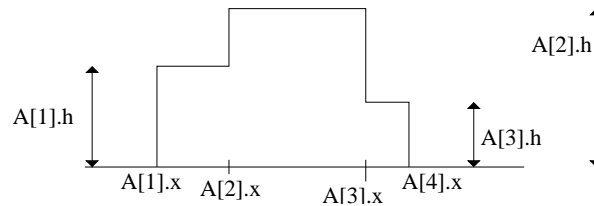
On décrit un bâtiment par un triplet  $(x_1, x_2, h) \in \mathbb{R}^3$  où  $x_1$  décrit la première coordonnée sur l'axe  $x$  et  $x_2$  la deuxième. La hauteur du bâtiment est décrite par  $h$ . Étant donné une liste de  $n$  bâtiments, la tâche est de trouver la silhouette (skyline) formée par nos bâtiments. Illustration:



La silhouette est représentée par la liste croissante de coordonnées  $x$  et la hauteur correspondante.

- a) Etant donné deux skylines on veut écrire un programme pour les combiner (c'est-à-dire trouver le skyline obtenu avec tous les bâtiments du premier et du deuxième).

On suppose qu'on nous donne deux tableaux  $A$  et  $B$  dont chaque entrée contient deux valeurs:  $x$  et  $h$ . Ces tableaux décrivent les deux skylines qu'on veut combiner. Le output est aussi un skyline, c'est-à-dire un tableau  $C$  avec le même type d'entrées, qui décrit le skyline obtenu en combinant les skylines de  $A$  et  $B$ .



Ecrire un algorithme qui, étant donné comme input les tableaux  $A$  et  $B$ , retourne comme output le tableau  $C$ . (*Indication:* Regarder MergeSort).

- b) Utiliser le fait qu'on puisse combiner deux skylines pour construire un algorithme diviser-pour-régner qui résout le problème Skyline en  $O(n \log(n))$ .