

# IFT2010 H06 — Examen final

Miklós Csűrös

18 avril 2006

Aucune documentation n'est permise à l'exception d'une feuille de  $8\frac{1}{2}'' \times 11''$ .  
L'examen vaut 181 points.

**Répondez à toutes les questions dans les cahiers d'examen.**

## 0 Votre nom (1 point)

Écrivez votre nom et code permanent sur tous les cahiers soumis.

## 1 Récurrences (30 points)

Le logarithme itéré  $L(n) = \lg^* n$  est défini par

$$L(n) = \begin{cases} 0 & \text{si } n = 1; \\ L(\lfloor \lg n \rfloor) + 1 & \text{si } n > 1. \end{cases}$$

Quand  $f$  est une fonction quelconque, on va écrire donc  $f^*(n)$  pour la fonction défini par  $f^*(1) = 0$  et  $f^*(n) = f^*(\max\{\lfloor f(n) \rfloor, 1\}) + 1$ . Remplissez le tableau suivant (à partir de la deuxième rangée) en utilisant la notation asymptotique  $\Theta$ .

$f(n)$	$f^*$
$\lg(n)$	$\Theta(\lg^* n)$
$\sqrt{n}$	
$n - 2$	
$n/2$	
$n/3$	

Aucune justification n'est nécessaire.

## 2 Ensembles dynamiques (30 points)

Considérez les opérations suivantes sur une structure représentant l'ensemble  $\mathcal{S}$ .

- `membre( $x$ )` vérifie si  $x \in \mathcal{S}$
- `insérer( $x$ )` insère l'élément  $x$  dans  $\mathcal{S}$  — la structure correspond à l'ensemble  $\mathcal{S} \cup \{x\}$  après l'exécution. Ici, on est certain que  $x \notin \mathcal{S}$  avant l'exécution.
- `supprimer( $x$ )` supprime l'élément  $x$  dans  $\mathcal{S}$  — la structure correspond à l'ensemble  $\mathcal{S} \setminus \{x\}$  après l'exécution. Ici, l'élément  $x$  est spécifié directement dans la structure et donc il n'est pas nécessaire de rechercher. (Par exemple, dans une liste chaînée, on utilise un pointeur à l'objet qu'on veut supprimer.)

Déterminez le temps de calcul, **dans le pire des cas**, de ces opérations dans l'implantation usuelle des structures suivantes. Donnez vos réponses en notation asymptotique pour un ensemble de taille  $n = |\mathcal{S}|$ . Aucune justification n'est nécessaire.

<b>structure</b>	<b>membre</b>	<b>insérer</b>	<b>supprimer</b>
liste chaînée			
arbre AVL			
tas binaire			
tableau de hachage avec sondage linéaire			

## 3 Hello moto (30 points)

Vous devez écrire un logiciel pour suivre les transactions de la Bourse sur un téléphone cellulaire. Le fournisseur transmet immédiatement les transactions et le logiciel doit afficher à tout temps la liste des  $m$  transactions les plus volumineuses de la journée. (Si  $v_1, v_2, \dots$  est la liste des valeurs transmises, on veut être capable d'énumérer les  $m$  valeurs qui sont les plus grandes parmi  $v_1, v_2, \dots, v_i$ , pour tout  $i$ .) Proposez une solution efficace avec l'analyse de son temps et espace de calcul. Comme il s'agit d'un cellulaire, il est essentiel de minimiser le temps et l'espace requis. En particulier, il est impossible de stocker la liste de toutes les transactions de la journée.

## 4 Adressage ouvert (30 points)

- Expliquez le phénomène de grappe forte (*primary clustering*).
- Expliquez le phénomène de grappe faible (*secondary clustering*).

c. On utilise un tableau de hachage de taille  $M = 2^{20}$  avec la méthode de la multiplication. Pour la résolution de collisions, on veut utiliser l'adressage ouvert. Un collègue a proposé la procédure de sondage suivante :

1. Pour placer un élément à clé  $x$ , calculer la valeur de la fonction de hachage  $h(x) \in \{0, 1, \dots, M - 1\}$ .
2. Mettre l'élément dans la première case vide de la série  $h_0, h_1, \dots$  où

$$\begin{aligned}h_0 &= h(x), h_1 = (h_0 + 1024) \bmod M, h_2 = (h_0 - 1024) \bmod M, \\h_3 &= (h_0 + 2048) \bmod M, h_4 = (h_0 - 2048) \bmod M, \\h_5 &= (h_0 + 3072) \bmod M, h_6 = (h_0 - 3072) \bmod M, \dots\end{aligned}$$

Dans d'autres mots,  $h_{2k} = (h_0 - 1024k) \bmod M$  et  $h_{2k+1} = (h_0 + 1024k) \bmod M$  pour tout  $k = 0, 1, \dots$

Est-ce que cette procédure cause des problèmes de grappe forte ou faible ? Est-ce qu'il y a d'autres fautes avec la procédure proposée ? Justifiez vos réponses.

## 5 Amortisation (30 points)

Expliquez la notion de *coût amorti* d'une opération.

## 6 Les convertisseurs (30 points)

On doit travailler avec des fichiers graphiques qui ont des formats différents (GIF, BMP, SVG, ...). Il existe  $n$  formats et pour chaque paire de formats  $(i, j)$  il existe un logiciel de conversion  $i \leftrightarrow j$  qui coûte  $p_{ij}$  dollars et effectue la conversion en un temps  $t_{ij}$  (disons, pour une image de taille «standard»). Votre employeur préfère le format SVG, et donc on veut être capable de performer la conversion de n'importe quel format vers SVG.

- Décrivez une méthode efficace pour calculer l'ensemble de logiciels à acheter si on veut effectuer les conversions le plus vite possible.
- Décrivez une méthode efficace pour calculer l'ensemble de logiciels à acheter si on veut minimiser le coût total des achats.