

# IFT2015 hiver 2012 — Devoir 4

Miklós Csűrös

18 septembre 2012

À remettre avant 20 :15 le 25 septembre. Faites **deux des trois** exercices pour avoir le maximum de 30 points. Vous avez le droit de soumettre une solution au troisième pour 15 points boni. Remettez un rapport écrit par courriel (à csuros@iro...) en format PDF. Travaillez seul.

## 4.1 Déploiement (15 points)

**On splaye à la main (5 points).** Dessiner l'ABR complet de hauteur 3 avec clés  $\{1, 2, \dots, 15\}$ . Montrer le résultat du déploiement (*splaying*) après chaque opération dans la séquence `search(12)`, `search(1)`, `delete(8)`.

**Séquence alternante (10 points).** Considérez la séquence d'opérations

`insert(1)`, `insert( $n$ )`, `insert(2)`, `insert( $n - 1$ )`, `insert(3)`,  $\dots$ , `insert( $\lceil (n + 1)/2 \rceil$ )`,

exécutées dans un arbre *splay* initialement vide. Démontrez que (1) le coût amorti des opérations est  $o(\log n)$  dans cette séquence, et que (2) la hauteur de l'arbre est  $\Theta(i)$  après le  $i$ -ème insertion. **Indice** : observez comment la structure de l'arbre change avec les opérations, énoncez un lemme, et démontrez-le par induction.

*Morale* : Le coût amorti  $O(\log n)$  d'un arbre *splay* n'implique ni que le coût amorti soit  $\Theta(\log n)$ , ni que la hauteur de l'arbre soit  $O(\log n)$ .

## 4.2 TAD Partition (15 points)

Proposez une structure efficace pour implanter un type abstrait qui représente la partition des nombres non-négatifs en intervalles<sup>1</sup>. Une partition est l'ensemble

---

<sup>1</sup>L'intervalle  $[a, b)$  est l'ensemble de valeurs réelles  $\{x : a \leq x < b\}$  où  $a \leq b$ . Il est permis d'avoir  $b = \infty$  : c'est interprété comme  $[a, \infty) = \{x : a \leq x\}$ . Notez que  $[a, a)$  est l'ensemble vide.

$\{[0, x_1), [x_1, x_2), \dots, [x_m, \infty)\}$  avec  $x_0 = 0 < x_1 < x_2 < \dots < x_m$  et un  $m = 0, 1 \dots$ , quelconque. L'interface du type comprend les opérations suivantes avec un paramètre  $x > 0$  :

recherche( $x$ ) retourne l'intervalle  $[x_{i-1}, x_i)$  qui contient  $x$   
 fusion( $x$ ) recherche  $x \in [x_{k-1}, x_k)$ , et remplace les intervalles  $[x_{k-1}, x_k)$  et  $[x_k, x_{k+1})$  par  $[x_{k-1}, x_{k+1})$   
 tranche( $x$ ) recherche  $x \in [x_{k-1}, x_k)$ , et remplace l'intervalle  $[x_{k-1}, x_k)$  par  $[x_{k-1}, x)$  et  $[x, x_k)$

Votre structure doit supporter les trois opérations en temps  $O(\log m)$ .



### 4.3 Recherche optimale (15 points)

*Cet exercice caractérise les ABRs minimisant le nombre de comparaisons arithmétiques dans le pire cas de la recherche fructueuse. Allant peut-être à l'encontre de l'intuition première, ce type d'arbres optimaux ne sont pas les arbres binaires complets.*

Soit  $t[N]$  le nombre de comparaisons arithmétiques dans l'algorithme suivant finissant avec nœud  $N$  (dernière comparaison de la recherche fructueuse :  $x = N.\text{key}$ ).

```

SEARCH( $x, N$ )           // recherche dans le sous-arbre du nœud  $N$ 
S1 if  $N = \text{null}$  then return null           // recherche infructueuse
S2 if  $x < N.\text{key}$  then return SEARCH( $x, N.\text{left}$ )
S3 if  $x = N.\text{key}$  then return  $N$            // recherche fructueuse
S4 else return SEARCH( $x, N.\text{right}$ )           //  $\{x > N.\text{key}\}$ 
    
```

**(a) Définition (3 points)** ► Donnez une *définition récursive* de  $t[N]$ . **Indice** : Il y a une comparaison à compter en Ligne S2 et une autre en Ligne S3. Notez que  $t$  croît différemment vers la gauche et vers la droite.

**(b) Isobathes (4 points)** Rappel :  $F(k)$  dénote le  $k$ -ème nombre Fibonacci :  $F(0) = 0$ ,  $F(1) = 1$  et  $F(k) = F(k-1) + F(k-2)$  pour tout  $k > 1$ . Soit  $T[k]$  le nombre de nœuds internes avec  $t[N] = k$ . ► Démontrez que pour tout  $k = 2, 3, \dots$ ,  $N[k] \leq F(k-1)$ . **Indice** : examinez comment la récurrence en (a) mène à une inégalité récursive pour  $N[k]$ .

**(c) Arbre optimal (4 points)** Soit  $t_{\max}$  le maximum de  $t[N]$  dans l'arbre. ► Caractériser les arbres avec le maximum nombre de nœuds internes pour un  $t_{\max}$  donné. **Indice** : l'arbre maximal n'est pas un arbre binaire complet mais il a une structure récursive très bien définie. . .

**(d) Le vrai nombre de comparaisons (4 points)** Soit  $\phi = \frac{1+\sqrt{5}}{2}$ . ► Démontrez que dans un arbre avec  $n$  nœuds internes,

$$t_{\max} \geq \log_{\phi}(n + 1) + c \quad (4.3.1)$$

où la constante  $c$  qui ne dépend pas de  $n$ . ► Comparez la borne de (4.3.1) avec le nombre de nœuds dans l'arbre maximal en (c). **Indice** : écrire  $n = N[t_{\max}] + N[t_{\max} - 1] + N[t_{\max} - 2] + \dots$  et utiliser (b)