# 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  $\{1, 2, ..., 15\}$ . Montrer le résultat du déploiement (*splaying*) après chaque opération dans la séquence search(12), search(1), delete(8).

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

$$\mathsf{insert}(1), \mathsf{insert}(n), \mathsf{insert}(2), \mathsf{insert}(n-1), \mathsf{insert}(3), \dots, \mathsf{insert}\Big( \left\lceil (n+1)/2 \right\rceil \Big),$$

executées 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

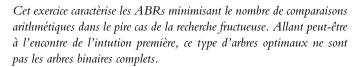
<sup>&</sup>lt;sup>1</sup>L'intervalle [a,b) est l'ensemble de valeurs réelles  $\{x\colon a\le x< b\}$  où  $a\le b$ . Il est permis d'avoir  $b=\infty$ : c'est interpreté comme  $[a,\infty)=\{x\colon a\le x\}$ . Notez que [a,a) est l'ensemble vide.

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

```
\begin{array}{ll} \operatorname{recherche}(x) & \operatorname{retourne l'intervalle}\left[x_{i-1},x_i\right) \operatorname{qui} \operatorname{contient} x \\ \operatorname{fusion}(x) & \operatorname{recherche} x \in [x_{k-1},x_k), \operatorname{et} \\ & \operatorname{remplace} \operatorname{les intervalles}\left[x_{k-1},x_k\right) \operatorname{et}\left[x_k,x_{k+1}\right) \operatorname{par}\left[x_{k-1},x_{k+1}\right) \\ \operatorname{tranche}(x) & \operatorname{recherche} x \in [x_{k-1},x_k), \operatorname{et} \\ & \operatorname{remplace} \operatorname{l'intervalle}\left[x_{k-1},x_k\right) \operatorname{par}\left[x_{k-1},x\right) \operatorname{et}\left[x,x_k\right) \end{array}
```

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

# 4.3 Recherche optimale (15 points)



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

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

- (a) Définition (3 points)  $\blacktriangleright$  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)** Rapel : 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.  $\blacktriangleright$  Démontrez que pour tout  $k = 2, 3, \ldots, N[k] \le F(k-1)$ . **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.  $\blacktriangleright$  Caractérisez 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  $\phi = \frac{1+\sqrt{5}}{2}$ .  $\blacktriangleright$  Démontrez que dans un arbre avec n nœuds internes,

$$t_{\text{max}} \ge \log_{\phi}(n+1) + c \tag{4.3.1}$$

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