

IFT2015 Hiver 2010 — Devoir 1

Miklós Csűrös

27 janvier 2010

À remettre lors de la démonstration le 3 février. Remettez un rapport écrit. Un rapport soumis par courriel (à csuros@iro. . .) en format PDF vaut 3 points de boni.

1 Appartenance-union avec rang (20 points)

Au cours on a considéré l'idée de stocker la taille des arbres créés dans une structure de données pour le problème d'appartenance-union. Dans cet exercice, on examinera une autre approche qui utilise un *rang* associé à chaque élément. Le rang donne de l'information sur la hauteur de chaque arbre. En fait, c'est toujours une borne supérieure sur la hauteur¹ de x .

```
MAKESET( $x$ )
1  $\text{id}[x] \leftarrow x$ 
2  $\text{rang}[x] \leftarrow 0$ 

UNION( $x, y$ )
1  $p \leftarrow \text{FIND}(x)$ ;  $q \leftarrow \text{FIND}(y)$ 
2 si  $p \neq q$  alors
3   si  $\text{rang}[p] < \text{rang}[q]$ 
4   alors  $\text{id}[p] \leftarrow q$ 
5   sinon  $\text{id}[q] \leftarrow p$ ; si  $\text{rang}[p] = \text{rang}[q]$  alors  $\text{rang}[p] \leftarrow \text{rang}[p] + 1$ 

FIND( $x$ )
1 tandis que  $x \neq \text{id}[x]$  faire  $x \leftarrow \text{id}[x]$ 
2 retourner  $x$ 
```

¹ La hauteur d'un élément est défini dans l'arbre de pointeurs id comme la longueur maximale d'une série $y, \text{id}[y], \text{id}[\text{id}[y]], \dots$ qui finit par x .

Considérez les arbres créés par une série quelconque d'opérations `makeset`, `union` et `find`

$$\text{makeset}(1), \dots, \text{makeset}(n), \underbrace{\text{union/find}(\dots), \dots, \text{union/find}(\dots)}_{\text{autant de fois qu'on veut}}$$

sur un ensemble de n éléments.

a. (7 points) Démontrez que pour tout x , on a $\text{rang}[x] \leq \text{rang}[\text{id}[x]]$ avec inégalité stricte si et seulement si $x \neq \text{id}[x]$. Indice : preuve par induction dans le nombre d'opérations `union`.

b. (6 points) Démontrez que pour un arbre enraciné à x (donc $x = \text{id}[x]$), le nombre d'éléments dans l'arbre est supérieur ou égal à $2^{\text{rang}[x]}$. Indice : preuve par induction dans le nombre d'opérations `union`.

c. (7 points) Démontrez qu'une opération `find` prend $O(\log n)$ temps. Indice : utilisez les lemmes de a. et b.

2 Oh, la notation (10 points)

a. (5 points) Démontrez, en utilisant la définition de Θ , que

$$3n^3 + 2\lceil \lg(n+1) \rceil = \Theta(n^3).$$

b. (5 points) Est-ce que $O(n^3) \setminus O(n^2) = \Theta(n^3)$? (« $\setminus$ » dénote la soustraction sur des ensembles : $O(n^3) \setminus O(n^2) = \{f : f \in O(n^3) \text{ mais } f \notin O(n^2)\}$) Justifiez votre réponse.

3 Puissance (10 points de boni)

Donnez un algorithme pour calculer x^n où $x \in [0, \infty)$ et $n \in \{0, 1, 2, 3, \dots\}$ selon la récurrence

$$x^n = \begin{cases} x \cdot x^{n-1} & \{n \bmod k \neq 0 \text{ où } n = k\} \\ \left(x^{n/k}\right)^k & \{n = 2k, 3k, 4k, \dots\} \\ 1 & \{n = 0\} \end{cases}$$

Analysez le temps de calcul en fonction de n et $k = 2, 3, 4, \dots$