Université d'Evry Paris-Saclay **Année universitaire 2021-2022**
M2 Data Science

Preuve du Master Thorem

Énoncé :

Si on a la relation de récurrence suivante avec $a \geq 1$ et $b > 1$

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

avec $\frac{n}{b}$ signifiant $\lfloor \frac{n}{b} \rfloor$ ou $\lceil \frac{n}{b} \rceil$ ALORS
1) Si $f(n) = O(n^{\log_b a - \epsilon})$ avec $\epsilon > 0$ alors $T(n) = O(n^{\log_b a})$
2) Si $f(n) = \Theta(n^{\log_b a})$ alors $T(n) = \Theta(n^{log_b a} \log(n))$
3) Si $f(n) = \Omega(n^{\log_b a + \epsilon})$ avec $\epsilon > 0$ et $af(\frac{n}{b}) \leq cf(n)$ avec $c < 1$ alors $T(n) = \Theta(f(n))$

On applique la relation de récurrence pour réduire n et faire disparaître la fonction T à droite et ne plus avoir une expression récurrente pour T.

On a :

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

d'où

$$T(n) = a\left(aT\left(\frac{n}{b^2}\right) + f\left(\frac{n}{b}\right)\right) + f(n)$$

et finalement

$$T(n) = \sum_{i=0}^{u} a^i f\left(\frac{n}{b^i}\right) + a^u \Theta(1)$$

avec $u = \lfloor \log_b(n) \rfloor$ (car $b^{\log_b(n)} = n$) et $\Theta(1)$ le temps des opérations élémentaires pour n petit (proche de 1).

$$a^{\log_b(n)} = a^{\log_a(n) \times \log_b(a)} = n^{\log_b(a)}$$

Ainsi :

$$a^u \Theta(1) = \Theta(a^{\log_b(n)}) = \Theta(n^{\log_b(a)})$$

CAS 1 avec $f(n) = O(n^{\log_b(a) - \epsilon})$

Pour n assez grand on a : $f(n) \leq Cn^{\log_b(a) - \epsilon}$ d'où

$$T(n) = \sum_{i=0}^{u} a^i f\left(\frac{n}{b^i}\right) + \Theta(n^{\log_b(a)}) \leq \sum_{i=0}^{u} C\left(a^i \left(\frac{n}{b^i}\right)^{\log_b(a) - \epsilon}\right) + \Theta(n^{\log_b(a)})$$

$$T(n) \leq C \sum_{i=0}^{u} \left(a^i \left(\frac{n}{b^i}\right)^{\log_b(a) - \epsilon}\right) + \Theta(n^{\log_b(a)})$$

$$T(n) \leq Cn^{\log_b(a)-\epsilon}\sum_{i=0}^{u}\left(a^i\left(b^{-i}\right)^{\log_b(a)-\epsilon}\right) + \Theta(n^{\log_b(a)})$$

or, $(b^{-i})^{\log_b(a)} = (b^{\log_b(a)})^{-i} = a^{-i}$, d'où

$$T(n) \leq Cn^{\log_b(a)-\epsilon}\sum_{i=0}^{u} b^{i\epsilon} + \Theta(n^{\log_b(a)})$$

$$T(n) \leq Cn^{\log_b(a)-\epsilon} \times \frac{b^{(u+1)\epsilon}-1}{b^\epsilon - 1} + \Theta(n^{\log_b(a)})$$

$$T(n) \leq Cn^{\log_b(a)-\epsilon} \times b^{\log_b(n)\epsilon}\frac{b^\epsilon}{b^\epsilon - 1} + \Theta(n^{\log_b(a)})$$

$$T(n) \leq Cn^{\log_b(a)-\epsilon} \times n^{\epsilon}\frac{b^\epsilon}{b^\epsilon - 1} + \Theta(n^{\log_b(a)})$$

$$T(n) \leq Cn^{\log_b(a)} \times \frac{b^\epsilon}{b^\epsilon - 1} + \Theta(n^{\log_b(a)})$$

ainsi on obtient bien $T(n) = O(n^{\log_b(a)})$

CAS 2 avec $f(n) = \Theta(n^{\log_b(a)})$

Pour n assez grand on a : $C_1 n^{\log_b(a)} \leq f(n) \leq C_2 n^{\log_b(a)}$ d'où

$$\sum_{i=0}^{u} C_1\left(a^i\left(\frac{n}{b^i}\right)^{\log_b(a)}\right) + \Theta(n^{\log_b(a)}) \leq T(n) \leq \sum_{i=0}^{u} C_2\left(a^i\left(\frac{n}{b^i}\right)^{\log_b(a)}\right) + \Theta(n^{\log_b(a)})$$

et donc

$$C_1 n^{\log_b(a)}\sum_{i=0}^{u} 1 + \Theta(n^{\log_b(a)}) \leq T(n) \leq C_2 n^{\log_b(a)}\sum_{i=0}^{u} 1 + \Theta(n^{\log_b(a)})$$

$$C_1 n^{\log_b(a)}(\lfloor \log_b(n)\rfloor + 1) + \Theta(n^{\log_b(a)}) \leq T(n) \leq C_2 n^{\log_b(a)}(\lfloor \log_b(n)\rfloor + 1) + \Theta(n^{\log_b(a)})$$

ce qui donne

$$T(n) = \Theta(\log_b(n) n^{\log_b(a)}) = \Theta(\log(n) n^{\log_b(a)})$$

CAS 3 avec $f(n) = \Omega(n^{\log_b a + \epsilon})$

Pour n assez grand on a : $Cn^{\log_b(a)+\epsilon} \leq f(n)$ d'où

$$C\sum_{i=0}^{u}\left(a^i\left(\frac{n}{b^i}\right)^{\log_b(a)+\epsilon}\right) + \Theta(n^{\log_b(a)}) \leq T(n)$$

et en reprenant les calculs effectués dans le cas 1 en remplaçant ϵ par $-\epsilon$:

$$Cn^{\log_b(a)+\epsilon} \times \frac{1-b^{-(u+1)\epsilon}}{1-b^{-\epsilon}} + \Theta(n^{\log_b(a)}) \leq T(n)$$

ou bien avec $D = C/(1-b^{-\epsilon})$:

$$Dn^{\log_b(a)+\epsilon}(1-b^{-(u+1)\epsilon}) + \Theta(n^{\log_b(a)}) \leq T(n)$$

$b^{-(u+1)\epsilon} = \left(\frac{1}{bn}\right)^{\epsilon}$ s'approche de 0 pour n grand, ainsi, pour n assez grand on aura :

$$\frac{D}{2}n^{\log_b(a)+\epsilon} + \Theta(n^{\log_b(a)}) \leq Dn^{\log_b(a)+\epsilon}(1-b^{-(u+1)\epsilon}) + \Theta(n^{\log_b(a)}) \leq T(n)$$

d'où

$$T(n) = \Omega(n^{\log_b(a)+\epsilon}) = \Omega(f(n))$$

Pour la majoration, on utilise l'hypothèse $af\left(\frac{n}{b}\right) \leq cf(n)$ qui donne par récurrence : $a^i f\left(\frac{n}{b^i}\right) \leq c^i f(n)$

Ainsi, sachant que $0 < c < 1$:

$$T(n) = \sum_{i=0}^{u} a^i f\left(\frac{n}{b^i}\right) + \Theta(n^{\log_b(a)}) \leq \sum_{i=0}^{u} c^i f(n) + \Theta(n^{\log_b(a)}) = f(n)\frac{1-c^{u+1}}{1-c} + \Theta(n^{\log_b(a)})$$

$$T(n) \leq f(n)\frac{1}{1-c} + \Theta(n^{\log_b(a)})$$

Or, dans ce cas 3, $f(n)$ croit plus rapidement que $n^{\log_b a}$ donc

$$T(n) = O(f(n))$$

Et on conclut bien que

$$T(n) = \Theta(f(n))$$

$\square$