

Master Theorem

"Théorème général" dans la langue de Charles AZNAVOUR

Clarence KINEIDER

Leçons : 903, 926, 931

Référence(s) : BEAUQUIER, BERSTEL, CHRÉTIENNE, *Éléments d'algorithmique*

Théorème : Soit $T : \mathbf{N} \rightarrow \mathbf{R}_+$ croissante à partir d'un certain rang. On suppose qu'il existe un rang $n_0 \geq 1$, des entiers $b \geq 2$ et $k \geq 0$ et des réels $a, c, d > 0$ tels que la relation de récurrence suivante soit vérifiée :

$$\begin{cases} T(n_0) = d \\ T(n) = aT\left(\frac{n}{b}\right) + cn^k \end{cases} \quad \text{pour tout entier de la forme } n = b^p n_0 \text{ avec } p \in \mathbf{N}$$

Alors on a la disjonction de cas suivante :

- Si $a < b^k$ alors $T(n) = \Theta(n^k)$
- Si $a = b^k$ alors $T(n) = \Theta(n^k \log_b(n))$
- Si $a > b^k$ alors $T(n) = \Theta(n^{\log_b(a)})$

Démonstration : Soit $p \in \mathbf{N}$, $n = b^p n_0$. Par récurrence, on a :

$$T(n) = da^p + \sum_{i=0}^{p-1} ca^i \left(\frac{n}{b^i}\right)^k = \underbrace{da^p}_{\delta(n)} + cn^k \underbrace{\sum_{i=0}^{p-1} \left(\frac{a}{b^k}\right)^i}_{\gamma(n)}$$

On a $\delta(n) = da^p = da^{\log_b(n) - \log_b(n_0)} = \frac{d}{a^{\log_b(n_0)}} b^{\log_b(a) \log_b(n)} = \frac{d}{a^{\log_b(n_0)}} n^{\log_b(a)} = \Theta(n^{\log_b(a)})$.

De plus, $\gamma(n)$ est une somme partielle d'une série géométrique. On distingue trois cas :

- Si $a < b^k$, la série géométrique de raison $\frac{a}{b^k}$ converge, donc $\gamma(n)$ est bornée. On a donc $T(n) = \Theta(n^{\log_b(a)}) + \Theta(n^k) = \Theta(n^k)$ car $\log_b(a) < k$.
- Si $a = b^k$, alors $\gamma(n) = p = \log_b(n) - \log_b(n_0) = \Theta(\log_b(n))$. Donc $T(n) = \Theta(n^{\log_b(a)}) + \Theta(n^k \log_b(n)) = \Theta(n^k \log_b(n))$ car $\log_b(a) = k$.
- Si $a > b^k$, alors $\gamma(n) \sim \alpha \frac{a^p}{b^{kp}}$ avec $\alpha = \frac{1}{\frac{a}{b^k} - 1}$. Donc $cn^k \gamma(n) \sim c\alpha \left(\frac{n}{b^p}\right)^k a^p \sim c\alpha n_0^k a^p$. Or $a^p = \Theta(n^{\log_b(a)})$, donc on a $T(n) = \Theta(n^{\log_b(a)}) + \Theta(n^{\log_b(a)}) = \Theta(n^{\log_b(a)})$.

On a donc le résultat pour n de la forme $n = b^p n_0$ avec $p \in \mathbf{N}$.

Soit $n \geq n_0$ et $p \in \mathbf{N}$ tel que $b^p n_0 \leq n < b^{p+1} n_0$. Puisque T est croissante à partir d'un certain rang, pour n assez grand on a $T(b^p n_0) \leq T(n) < T(b^{p+1} n_0)$. Or pour $f \in \{m \mapsto m^k, m \mapsto m^k \log_b(m), m \mapsto m^{\log_b(a)}\}$, on a $f(bm) = \Theta(f(m))$. Par encadrement, on a le résultat pour tout n à partir d'un certain rang. \square

Remarques :

- Suivant votre vitesse, vous pouvez ajouter une application à l'algorithme de votre choix (tri fusion pour la leçon 903 par exemple).
- Si on a seulement une majoration dans l'hypothèse de récurrence (i.e. $T(n) \leq aT\left(\frac{n}{b}\right) + cn^k$), on a le même résultat mais avec des O à la place des Θ . En effet, il suffit de voir qu'on a $T(n) \leq \delta(n) + cn^k \gamma(n)$ pour les mêmes δ et γ que dans la démonstration. Cela permet d'appliquer le théorème à des relations de type $T(n) = aT\left(\frac{n}{b}\right) + O(n^k)$.
- Voici une technique pour se passer de l'hypothèse de croissance de T pour le tri fusion :

On note $T(n)$ le temps d'exécution dans le cas le pire pour une entrée de taille n . On pose $T_m(n) = \max_{k \leq n} T(n)$ le temps d'exécution dans le cas le pire pour une entrée de taille inférieure ou égale à n . On a alors T_m croissante. On montre que T_m vérifie les hypothèses du Master Theorem avec une inégalité :

$$T_m(n) \leq T_m\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + T_m\left(\left\lceil \frac{n}{2} \right\rceil\right) + cn \leq 2T_m\left(\left\lceil \frac{n}{2} \right\rceil\right) + cn$$

Donc pour $n = 2^k$ on a :

$$T_m(2^k) \leq 2T_m(2^{k-1}) + cn$$

Par le Master Theorem, on a donc $T_m(n) = O(n \log(n))$ car T_m est croissante. Donc $T(n) \leq T_m(n) = O(n \log(n))$.