TD n° bonus 2

# *Diviser pour régner*

Hugo Salou MPI*

Dernière mise à jour le 20 mars 2023

# 1 Suites récurrentes de complexité

(*a*) On considère la suite $u_0 = 1$ et $u_n = u_{n-1} + a$ pour $a > 0$. Montrons que $u_n = \Theta(n)$.

(*b*) On considère la suite $u_0 = 1$ et $u_n = u_{n-1} + an$ pour $a > 0$.

(*c*) On considère la suite $u_0 = 1$ et $u_n = au_{n-1} + b$ pour $a > 2$ et $b \in \mathbb{R}_+^\star$.

$$\begin{aligned}
u_n &= au_{n-1} + b \\
&= a(au_{n-2} + b) + b \\
&= a^2 u_{n-2} + ab + b \\
&= a^3 u_{n-3} + a^2 b + ab + b \\
&\ \vdots \\
&= a_n + \sum_{k=0}^{n-1} a^k b \\
&= a^n + b\frac{1-a^n}{1-a} \\
&= a^n \cdot \left(1 - \frac{b}{1-a}\right) + \frac{b}{1-a} \\
&= \Theta(a^n)
\end{aligned}$$

(*d*) On considère la suite $u_0 = 1$ et $u_n = u_{n/2} + b$ avec $b > 0$. Soit $(v_p)_{p\in\mathbb{N}}$ la suite définie par $v_p = u_{2^p}$. Donc

$$v_p = u_{2^p} = u_{2^p/2} + b = u_{2^{p-1}/2} + 2b = \cdots = v_0 + bp = 1 + (p+1)b.$$

La suite $(u_n)_{n\in\mathbb{N}}$ est croissante donc, pour $n \in \mathbb{N}$, Alors,

$$\begin{aligned}
v_{\lfloor \log_2 n \rfloor} \leqslant u_n \leqslant v_{\lceil \log_2 n \rceil} \text{ d'où } & b(\lfloor \log_2 n \rfloor + 1) + 1 \leqslant u_n \leqslant b \\
\text{d'où } & b\log_2 n + 1 \leqslant u_n \leqslant b(\log_2 n + 2) + 1
\end{aligned}$$

On en déduit que $u_n = \Theta(\log_2 n)$.

(*e*) On considère la suite $u_0 = 1$ et $u_n = u_{n/2} + bn$ avec $b > 0$. On pose $(v_p)_{p\in\mathbb{N}}$ la suite définie par $v_p = u_{2^p}$.

$$\begin{aligned}
v_p &= v_{p-1} + 2^p b \\
&= v_{p-2} + 2^{p-1} b + 2^p b \\
&= b\sum_{k=1}^{p} 2^k + v_0 \\
&= b\sum_{k=0}^{p} 2^k + u_1 \\
&= b\sum_{k=0}^{p} 2^k + 1 + b \\
&= b\sum_{k=0}^{p} 2^k + 1 \\
&= b(2^{p+1} - 1) + 1.
\end{aligned}$$

D'où, par croissance de la suite $(u_n)_{n\in\mathbb{N}}$,

$b(2^{\log_2 n} - 1) + 1 \leqslant u_n \leqslant b(2^{\log_2 n + 2} - 1) + 1$ d'où $bn - b + 1 \leqslant u_n \leqslant 4bn - b + 1.$

Ainsi, $u_n = \Theta(n)$.

## 2 Multiplication d'entiers par algorithme de Karatsuba

1. On peut éventuellement rajouter des zéros à gauche jusqu'à avoir la même taille pour les deux nombres.
2. L'algorithme a une complexité en $\mathcal{O}(n^2)$.
3. On a $u = u_{\mathrm{a}} + 2^p \cdot u_{\mathrm{b}}$, où $p = |u|/2$. De même, $v = v_{\mathrm{a}} + 2^p \cdot v_{\mathrm{b}}$. On remarque que
$$\mathrm{prod}(u, b) = \mathrm{prod}(u_{\mathrm{a}}, v_{\mathrm{a}}) + 2^p\big(\mathrm{prod}(u_{\mathrm{a}}, v_{\mathrm{b}}) + \mathrm{prod}(u_{\mathrm{b}}, v_{\mathrm{a}})\big) + 2^{2p} \cdot \mathrm{prod}(u_{\mathrm{b}}, v_{\mathrm{b}}).$$
Avec $|u| = 2^p$, on note la complexité $C_p$. On a $C_p = 4C_{p-1} + K = 4(4C_{p-2} + K) + K = 4^p C_0 + K \sum_{i=0}^{p-1} 4^i$. On en déduit que $C_o = \Theta(4^p)$. Or, $p = \log_2 |u|$, d'où $4^{\log_2 |u|} = n^2$. L'algorithme est en $\mathcal{O}(n^2)$.
4. On remarque que $u_{\mathrm{a}}v_{\mathrm{a}} + u_{\mathrm{b}}v_{\mathrm{b}} - (u_{\mathrm{a}} - u_{\mathrm{b}})(v_{\mathrm{a}} - v_{\mathrm{b}}) = u_{\mathrm{a}}v_{\mathrm{a}} + u_{\mathrm{b}}v_{\mathrm{b}}$. Ainsi, $C_p = 3C_{p-1} + K$, d'où $C_p = \Theta(3^p)$. On en déduit que l'algorithme est en $\mathcal{O}(n^{\log_2 3})$.