03ª)

$$T(n) = \begin{cases} \Theta(1) & n = 1 \\ 3T(\frac{n}{3}) + \Theta(1) + n & n > 1 \end{cases}$$

Teorema mestre.

2º caso: Se $f(n) = \Theta(n^{\log_a^b})$, então $T(n) =$ $\Theta(n^{\log_a^b} \log n)$
p/ todo $a \geq 1$ e $b > 1$
constantes.

<u>Prova:</u>

$a = 3$ $\quad$ $0 \leq c_1 \cdot n^{\log_a^b} \leq n \leq c_2 \cdot n^{\log_a^b}$

$b = 3$

$f(n) = n$ $\quad$ $0 \leq n$, como $n > 1$

logo $0 \leq n$ //

$\boxed{\log_3^3 = 1}$ //

$c_1 \cdot n \leq n \quad (\div n)$

$c_1 \leq 1$

$\boxed{c_1 = 1}$ //

$n \leq c_2 \cdot n \quad (\div n)$

$1 \leq c_2$

$\boxed{c_2 = 2}$ //

Logo, $\forall n \geq 2$, $c_1 = 1$ e $c_2 = 2$.

Por sua vez, a complexidade é $\Theta(n \log n)$

Como já é sabido, a versão da merge sort padrão tem complexidade $\Theta(n \log n)$ também.

Concluo que não há diferença entre a merge padrão e a merge dividindo em 3, pois $\Theta(n \log n) = \Theta(n \log n)$.