

Algoritmo MergeSort (A, esq, dir)

Se  $esq < dir$  então  $\rightarrow \Theta(1)$

meio =  $(esq + dir) / 2 \rightarrow \Theta(1)$

MergeSort (A, esq, meio)  $\rightarrow T(\frac{n}{2})$

MergeSort (A, meio + 1, dir)  $\rightarrow T(\frac{n}{2})$

Intercala (A, esq, meio, dir)  $\rightarrow \Theta(n)$

Temos:

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n) + \Theta(1)$$

Reconstrução:

$$\begin{cases} T(n) = 2T\left(\frac{n}{2}\right) + n \\ T(1) = 1 \end{cases}$$

Atribuindo Valores:

$$T\left(\frac{n}{2}\right) = 2 \left[ 2T\left(\frac{n}{4}\right) + \frac{n}{2} \right] + n$$

$$T\left(\frac{n}{2}\right) = 4 \cdot T\left(\frac{n}{4}\right) + 2n$$

$$T\left(\frac{n}{2}\right) = 2^1 \cdot T\left(\frac{n}{2^2}\right) + 2n$$

⋮ em  $i$  iterações

$$T(n) = 2^i \cdot T\left(\frac{n}{2^i}\right) + in$$

$\Rightarrow$  Vai parar quando  $T\left(\frac{n}{2^i}\right) = 1$ , e

$T\left(\frac{n}{2^i}\right) = 1$ , se  $i = \log_2 n$ , logo:

$$\boxed{i = \lg n \text{ e } n = 2^i} \quad \therefore$$

$$T(n) = 2^{\lg n} \cdot T\left(\frac{n}{2^{\lg n}}\right) + n \lg n$$

$$T(n) = n \cdot T(1) + n \lg n$$

$$T(n) = n + n \lg n \Rightarrow T(n) = O(n \lg n) //$$