

We define  $T(n)$  as the time complexity of the merge sort  
 We define  $D(n)$  as the divide algorithm's time complexity  
 We define  $C(n)$  as the combine algorithm's time complexity

Thus

$$T(n) = \begin{cases} \Theta(n) & \text{if } n \leq c, \\ aT(n/2) + D(n) + C(n) & \text{otherwise.} \end{cases}$$

Because we are just computing  $\text{length}(a)$  :

$$D(n) = \Theta(1)$$

Because we are merging at max an array of  $n$  elements :

$$C(n) = \Theta(n)$$

$$T(n) = 2T(n/2) + cn$$

if  $n > 1$

$$\begin{aligned} T(n/2) &= 2T(n/4) + cn/2 \Rightarrow T(n) = 4T(n/4) + 2 * cn \\ &\Rightarrow T(n) = cn(\log(n) - 1) + 2 * cn \Rightarrow T(n) = cn\log(n) + cn \end{aligned}$$