

Q. Derive the worst case time complexity of Merge Sort using either tree method or substitution.

Ans: I'll use substitution method to solve the time complexity.

Merge sort algorithm contains two methods -

MergeSort(), which divides  
Merge(), which conquers and combines.

General time complexity of Merge sort is -

$$T(n) = \begin{cases} 1 & n=1 \\ 2T(n/2) + \theta(n) & n>1 \end{cases}$$

$$T(n) = 2T(n/2) + n \quad \text{--- (i)}$$

$$\text{if } T(n) = T(n/2) \Rightarrow$$

$$T(n/2) = 2T(n/4) + n/2 \quad \text{--- (ii)}$$

Now (ii) into (i)  $\Rightarrow$

$$T(n) = 2 \{ 2T(n/4) + n/2 \} + n$$

$$= 4T(n/4) + 2n \quad \text{--- (iii)}$$

if  $T(n) = T(n/4) \Rightarrow$

$$T(n/4) = 2T(n/8) + \frac{n}{4} \quad \text{--- (iv)}$$

Now (iv) into (iii)  $\Rightarrow$

$$\begin{aligned} T(n) &= 4 \left( 2T(n/8) + \frac{n}{4} \right) + 2n \\ &= 8T(n/8) + 3n \end{aligned}$$

$\vdots$

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

remember,  $T(1) = 1$

So, if  $\frac{n}{2^i} = 1$ , then we can say the time complexity.

$$\frac{n}{2^i} = 1$$

$$\Rightarrow n = 2^i$$

$$\Rightarrow \log_2 n = \log_2 2^i$$

$$\Rightarrow i = \log_2 n$$

So, worst time complexity of Merge Sort is ' $\log_2 n$ '