Time Complexity Analysis of MSS

Recurrence Relation of MSS Divide & conquer algorithm is

$$T(N) = 2T(N/2) + \Theta(N)$$

Let $\Theta(N) = N$

$$T(N) = 2T(N/2) + N \rightarrow (1)$$

By substitution Method.

$$T(N/2) = 2T(N/4) + N/2$$

Substitute this in (1)

$$T(N) = 2\left[2T(N/4) + N/2\right] + N$$

$$= 4T(N/4) + N + N \quad = 4T(N/4) + 2N \rightarrow (2)$$

$$T(N/4) = 2T(N/8) + N/4$$

Substitute in (2)

$$T(N) = 4\left[2T\left(N/8 + N/4\right)\right] + 2N$$

$$= 8T\left[N/8\right] + N + 2N$$

$$= 8T\left[N/8\right] + 3N$$

$$= 2^3 T\left[N/2^3\right] + 3N$$

$$\Rightarrow 2^K T\left[N/2^K\right] + KN$$

Let $N/2^K = 1$

$N = 2^K$

$K = \log_2 N$

$$\Rightarrow N \cdot T(1) + \log_2 N \cdot N$$

$$\Rightarrow T(N) = O(N\log N)$$

$\therefore$ The time complexity of MSS is $O(N\log N)$

By using Master Theorem

$T(N) = 2T(N/2) + O(N)$

$f(N) = O(N)$

$a = 2$

$b = 2.$

$g(N) = N^{\log_b^a}$

$= N^{\log_2^2}$

$= N$

$\Rightarrow O(N)$

$\because g(N) = f(n)$

$\Rightarrow T(n) = (N^{\log_b^a} \cdot \log N)$

$\Rightarrow T(N) = O(N \log N)$

$\therefore$ MSS time complexity is $O(N \log N)$