Iterative Approach:

An easy example

Let's say $T(n) = T(n-1) + 1$

$$T(n) = \begin{cases} 1 & n = 0 \\ T(n-1) + 1, & n > 0 \end{cases}$$

$$\therefore T(n) = T(n-1) + 1$$
$$= T(n-2) + 2$$
$$= [T(n-3) + 1] + 2$$
$$= T(n-3) + 3$$

~~will stop when~~

$$= T(n-k) + k$$

will stop when $n - k = 0$
$$\Rightarrow n = k$$

$$\therefore T(n) = T(n-n) + n$$
$$= T(0) + n$$
$$= 1 + n$$

$$\therefore T(n) = n$$

Merge Sort (Running Time)

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

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

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

$$= 4\left[2T(n/8) + \frac{n}{4}\right] + 2n$$

$$= 8T(n/8) + 3n$$

$$= 2^3 T\left(\frac{n}{2^3}\right) + 3n$$

general $= 2^k T\left(\frac{n}{2^k}\right) + kn$ ..... (1)

base case: $T(1) = O(1) = 1$

This expansion will go on till the number of element is 1.

$$\therefore \frac{n}{2^k} = 1$$

$$\Rightarrow n = 2^k$$

$$\Rightarrow k = \log_2 n$$

BRAC UNIVERSITY
Inspiring Excellence

Date:......................

that means, when $k = \log_2 n$

the iterative expansion will

stop.

$\therefore$ putting the value of k

in eq^n (1)

$$T(n) = 2^{\log_2 n} T(1) + n \log_2 n$$

$$= n^{\log_2 2} \cdot 1 + n \log_2 n$$

$$= n + n \log n$$

taking the max,

$$T(n) = n \log n$$

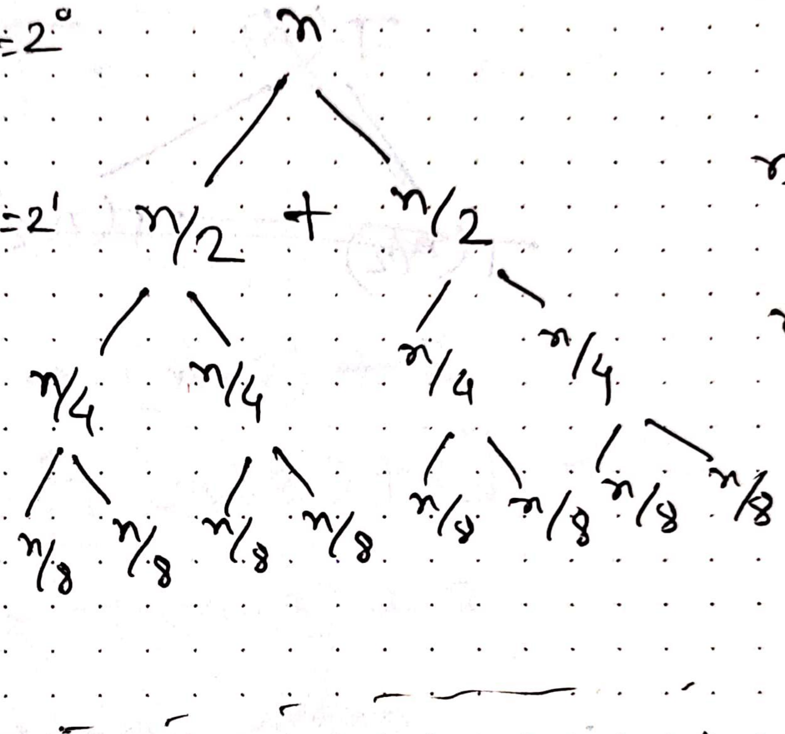
$\therefore$ Merge sort running

time is $n \log n$.

Recursive Tree Method:

$$T(n) = 2T(n/2) + n \;, \quad T(1) = 1$$

Recursive Call	#nodes	Tree	total operations
$T(n)$ $i=0$	$1 = 2^0$		n
$T(n/2)$ $i=1$	$2 = 2^1$		n
$T(n/4)$ $i=2$	$4 = 2^2$		n
$T(n/8)$ $i=3$	$8 = 2^3$		n
$T\left(\frac{n}{2^i}\right)$	2^i		n



+

$$= \sum_{i=0}^{\log_2 n} n$$

$$= n \sum_{i=0}^{\log_2 n}$$

$$= n\underbrace{(1+1+\cdots+1)}_{\log_2 n \text{ times}}$$

$$= n\log n$$

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

$$\Rightarrow i = \log_2 n$$

$$\therefore \boxed{T(n) = n\log n}$$

Date:....................

BRAC UNIVERSITY
Inspiring Excellence

Quick Sort - Running Time

* worst case happens when pivot is always the minimum / maximum.

* one partition is empty

* worst case running time of partitioning is $O(n)$

$\therefore T(n) = T(0) + T(n-1) + O(n)$

$= O(1) + T(n-1) + O(n)$

$= T(n-1) + 1 + n$

$= T(n-2) + 2 + 2n$

$= T(n-3) + 3 + 3n$

$= T(n-k) + k + kn$

will stop when $n-k = 0$

$\Rightarrow k = n$

$\therefore T(n) = T(0) + n + n^2$

$= 1 + n + n^2$

$$\boxed{T(n) = n^2}$$

Date:.......................

Recursive Tree method:

$$T(n) = T(0) + T(n-1) + cn$$

Recurrence	# of node	Tree	operation
$T(n)$	1	cn	cn
$T(n-1)$	2	$T(0)$ [1] $+ c(n-1)$	cn
$T(n-2)$	3	$T(0)$ [2] $c(n-2)$ [3]	cn
$T(n-3)$	4	$T(0)$ $c(n-3)$	cn
$T(n-k)$	$k+1$	$T(0)$ $c(n-k)$	cn

$\therefore$ $k+1$ levels

$$= (k+1)cn$$
$$= (n+1)cn$$
$$= cn^2 + cn$$

$n - k = 0$

$\Rightarrow k = n$

$$\therefore \boxed{T(n) = n^2}$$