

Q3 Explain how does merge sort algorithm use Divide and Conquer technique. Also analyze the time complexity.

Ans3

Merge sort is comparison based algorithm mainly it follows divide and conquer strategy.

Merge sort with the help of divide and conquer divides the array list into half halves until single element is left in the sublist. Then arrange them in ascending / descending order & combine the solutions of each sub list elements. After, applying merge sort →

Algorithm

while ($h \leq mid \ \&\& \ j \leq high$)
initially $i = low$

179, 285, 310, 652	284, 351, 413, 86
↑ h	↑ mid
	↑ j

if ($a[h] \leq a[j]$)

$b[i] = a[h];$
 $h++;$

assigning $a[h]$ to $b[i]$ /
increment it swapping

else { // other condition

$b[i] = a[j];$
 $j++;$

→ copying of value

$i++;$

Answer store $\rightarrow 179, 254, 285, 310, 351, 423, 652, 861$

when while condition fails then,

if ($h > mid$) \rightarrow 1st array list complete

```

{
  for (k = j to high) do
  {
    b[i] = a[k];
    i++;
  }
}

```

\rightarrow copy all the values in array remaining

else \rightarrow 2nd array list complete

```

{
  for (k = h to mid) do
  {
    b[i] = a[k];
    i++;
  }
}

```

\rightarrow copy all the values in array remaining

Time Complexity

Recurrence relation

$$T_n = \begin{cases} a & n=1 \\ 2T\left(\frac{n}{2}\right) + cn & n > 1 \end{cases}$$

Calculation

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

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

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

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

$$= 4T\frac{n}{4} + cn + cn$$

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

$$= 8T\frac{n}{8} + cn + 2cn$$

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

$$= 2^k \times T\left(\frac{n}{2^k}\right) + kcn$$

Assume $\underline{n = 2^k}$

$$= n \times T(1) + \log_2 n \cdot cn$$

$$= n \cdot 1 + cn \log_2 n$$

$$= O(n \log_2 n)$$