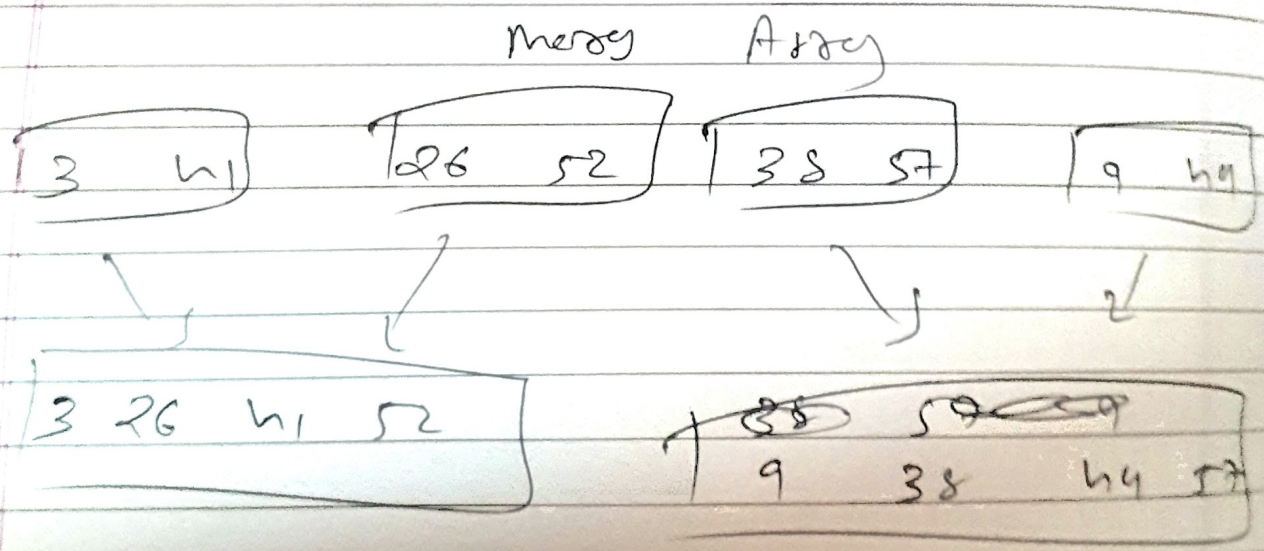
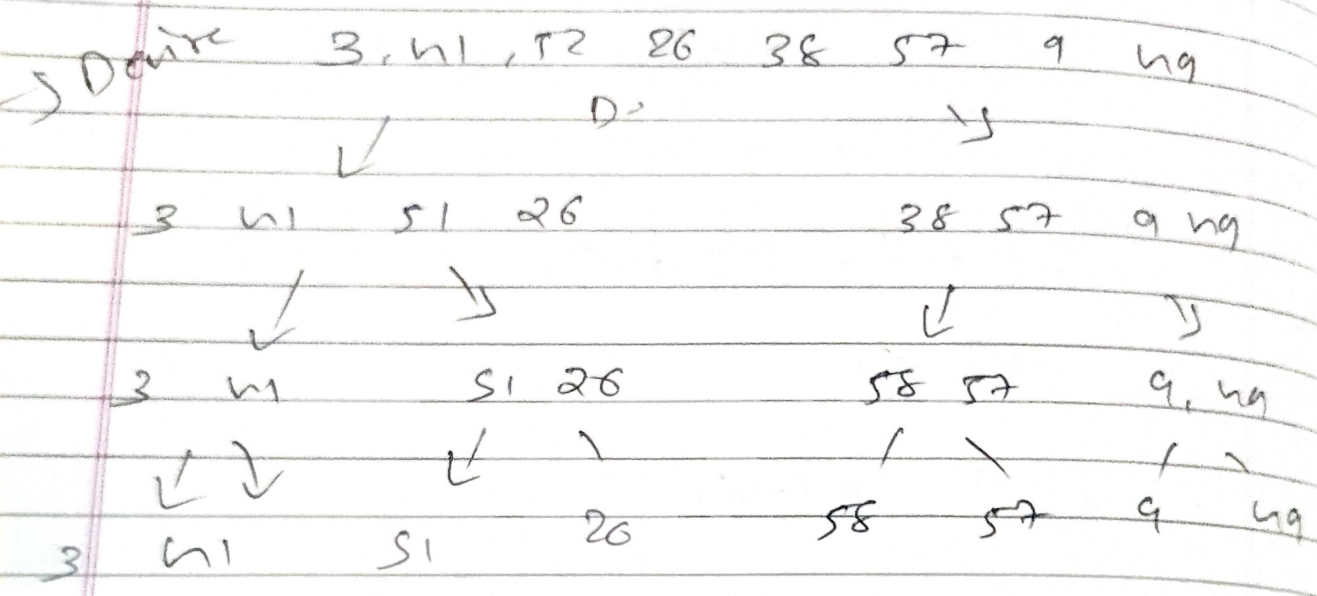


Q:1 A) merge sort

A: 3, 41, 52, 26, 38, 57, 9, 49



Final Array

3, 9, 26, 38, 41, 49, 52, 57

271

B →

$$T(m) \approx 2T(m/2) + m.$$

Q 22

A) $2^{n+1} = O(2^n)$

for

C and $n > 0$

$$0 \leq 2^{n+1} \leq C \cdot 2^n \quad (\text{for } n), n > 0$$

$$2^{n+1} = 2 \times 2^n$$

$$C = 2 \quad A = 1$$

so $2^{n+1} = O(2^n)$

B

$$2^{2n} = O(2^n)$$

for c and $n > 0$

$$0 \leq 2^{2n} < c \cdot 2^n$$

$$2^{21} < 2^{21}$$

$$2^n \times 2^n \leq c \times 2^n$$

$$2^n \not\leq c$$

$$2^{2n} \neq O(2^n)$$

if $T(n)$ is $O(\lg n)$

then

$T(n) \leq c \lg n$ for constant c and all $n > n_0$

so

$$T(n/2) \leq c \lg(n/2)$$

$$\begin{aligned} T(n) &= T(n/2) + 1 \\ &\leq c \lg(n/2) + 1 \\ &\leq c \lg n - c + 1 \\ &\leq c \lg n - c + 1 \\ &\leq c \lg n \end{aligned}$$

$c \geq 1$

so ~~$c \geq 1$~~

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

where $a=1$ and $b=2$

$$f(n) = n^2$$

$$= n \log^q n = n^1 = n^2$$

Case 2 =

$$T(n) = O(n^2 \log n)$$

$$O(n) = \underline{\underline{B}}$$