

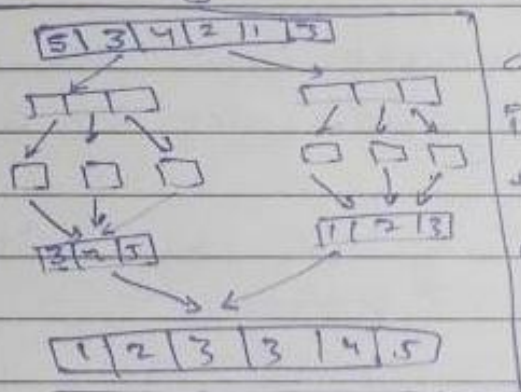
## Assignment - (2)

Answer 1)

(a) as we have to custom data structure DSA supporting two operations 'get-at-index (j)' or 'set-at-index (j, x)' in  $O(\log K)$

→ Merge Sort is most effective for this case and it uses the concept of divide & conquer

↳ Algorithm →



Sorting recursively by dividing the array into two equal halves then sorting each halves and then merging back together.

→ Since, it has better Asymptotic time comp.  $O(n \log n)$  compared to insertion sort ( $O(n^2)$ ) & selection sort ( $O(n^2)$ )

→ It is also noticed that in merge sort main op's are comparison using swaps

→ ∴ the Merge Sort provides the low time complexity even through it requires more additional space

Ans  
(b)

$A \rightarrow$  array  $\leftarrow$  fixed  
size of ' $n$ ' integers / comparable objects

for this case, Again the Merge sort

$\rightarrow$  principle divide & conquer

$\rightarrow$  doesn't rely on swapping elements directly but rather on merging sorted subarrays.

$\rightarrow$  the  $O(\log n)$  time for comparing objects is align same as "Merge step" of Merge sort

$\rightarrow$  And Merge Sort's worst case time complexity of  $O(n \log n)$  make it a better choice.

when we comparing ' $n$ ' obj. with  $O(\log n)$ . for  $n$

$\rightarrow$  therefore Merge sort is a suitable choice for this scenario due to its stable time complexity and ease of implementation.



Answer (c) given sorted array 'A' containing 'n' integers

- swapping  $\log \log n$  btw adjacent items
  - Insertion Sort: → time complexity  $O(n^2)$  in worst case which is not suitable for this case because due to the large no. of swaps
  - Selection Sort: → time complexity  $O(n^2)$  in worst case, making it not suitable for efficiently handling the swaps
  - Merge Sort: → time complexity  $O(n \log n)$  in all cases. It can handle swaps efficiently without increasing the time complexity.
- ∴ Merge Sort is the best choice for re-sorting the integers in array 'A' after  $\log \log n$  swaps btw adjacent time.