

Q1 -)

a) for in place sorting,

Insertion, selection sort is preferred.  
merge sort is not an in-place sorting algorithm.

The ~~best~~ worst case time-complexity  
of ~~no comparisons~~ <sup>Swap</sup> (ie set-at) in  
insertion is  $O(n^2)$

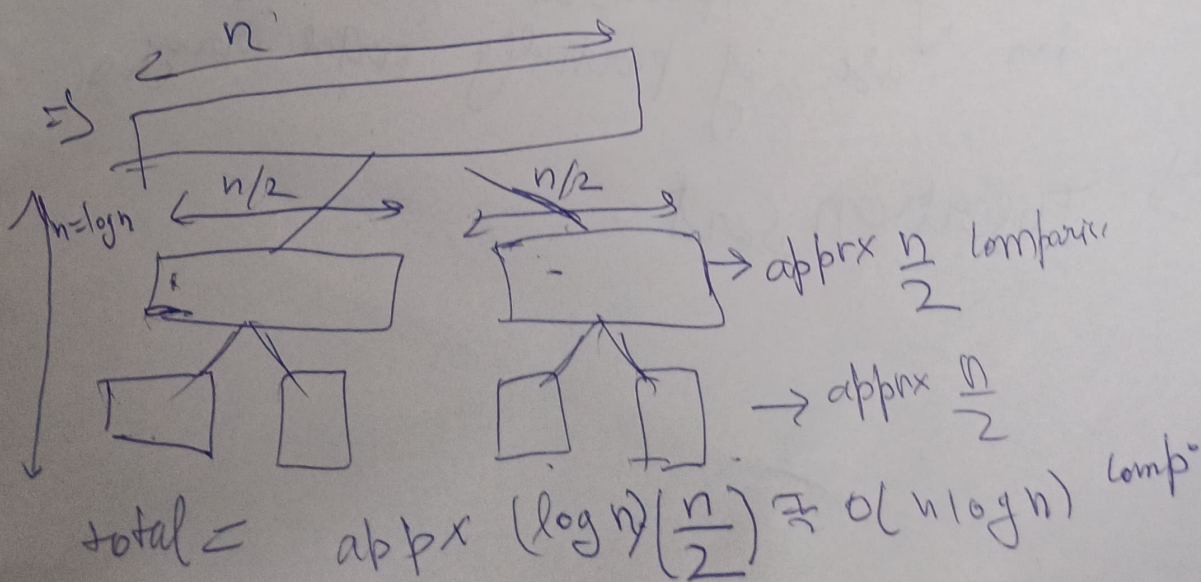
but in selection swap (set-at) is  $O(n)$

So, swaps in selection < swaps in insertion

⇒ Selection sort is preferred

b) comparisons in merge sort <

comparisons in selection sort  
and comparison in insertion  
sort (worst case)



in selection sort comparison  $\approx O(n^2)$  (worst case)  
Selection sort comparison  $\approx O(n^2)$ .

$\Rightarrow$  ~~but~~  $O(n \log n) < O(n^2)$

$\Rightarrow$  merge sort

(\*) we need to perform  $\log \log n$   
swaps to make array sort  
which is possible in insertion sort.  
insertion sort is the best choice  
for partially sorted array because  
it is based on no. of inversions.

while selection & merge do not  
depend on no. of inversion because  
in these treat partially sorted  
as worst case and take larger  
no. of comparison than insertion  
in case of partially sorted array.

2) Insertion Sort -