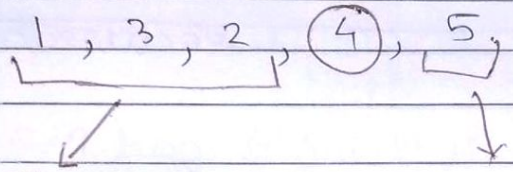
Quick Sort

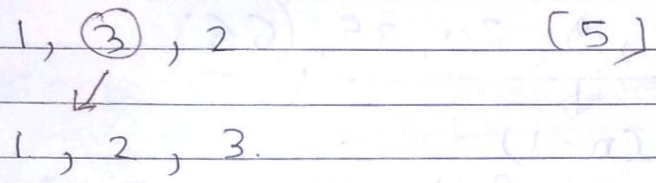
Page No.
Date :

* Pivot :- Choose any element → after 1st pass all the elements < P will be on LHS of P & elements > P will be at RHS of P

Ex :- 5, (4), 3, 2, 1



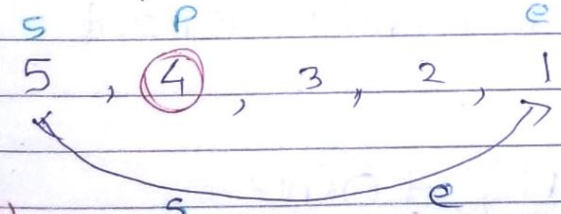
So after every pass you are putting pivot at the right position



1, 2, 3.

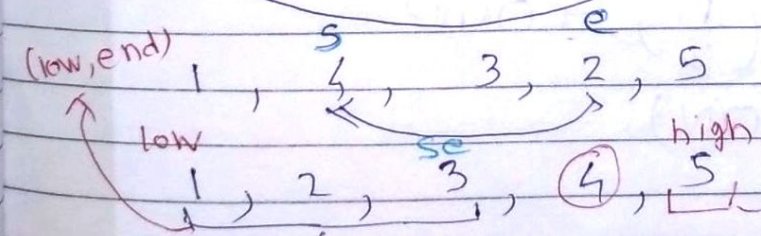
* In the merge sort when the array was sorted then also it will go till the very end, till the base condition. Here that will not happen

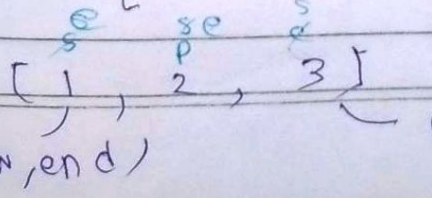
• How to pivot at correct position ?



P = 4.

while n[s] < P
s++
while n[e] > P
~~e++~~ e--

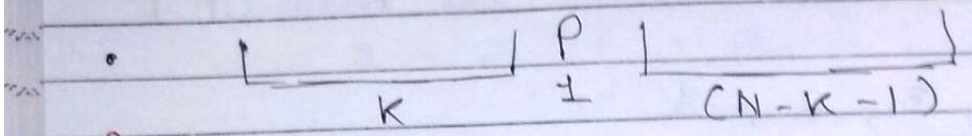




s & end → for swapping
low & high → tells us which part of the array you are working on

Page No.
Date :

* How to pick pivot?:-
  - Random element
  - Corner element
  - pick the middle element

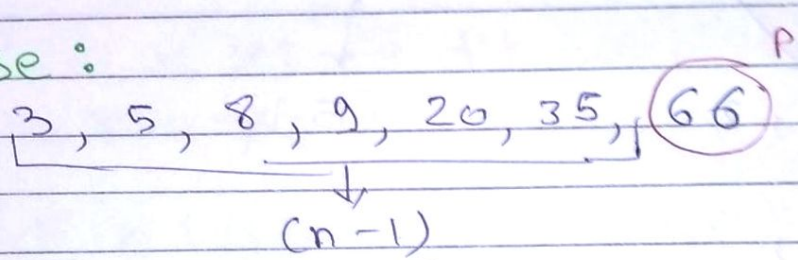


Imp

$$T(N) = T(K) + T(N-K-1) + O(N)$$

↳ recurrence relation

worst case :



$K = 0$ (when one part of the array is empty)

$$T(N) = T(0) + T(N-1) + O(N)$$

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

$$= O(N^2)$$

Best case : when pivot is middle element

$$K = \frac{N}{2}$$

$$T(N) = T\left(\frac{N}{2}\right) + T\left(\frac{N}{2}\right) + O(N)$$

$$T(N) = 2T\left(\frac{N}{2}\right) + O(N)$$

$$= O(N \log N)$$

Page No:
Date :

Notes :-

* Not Stable Already

* In-place i.e. why preferred for arrays instead of merge sort. MS. takes O(N) extra space.

* MS is better in linked list due to memory allocation

* Hybrid Sorting Algorithm (Tim Sort):

Merge Sort + Insertion sort,

↓

works well with partially sorted data