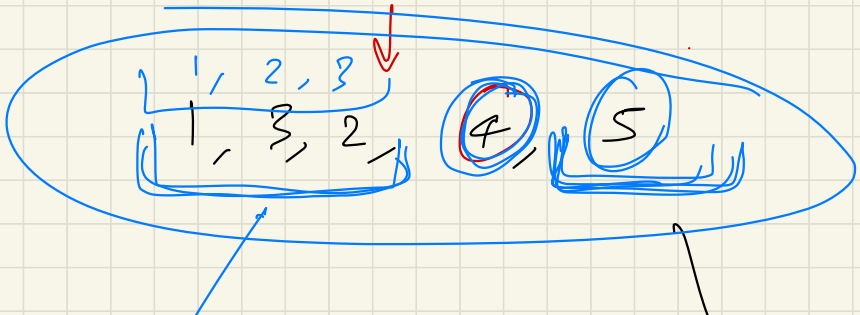
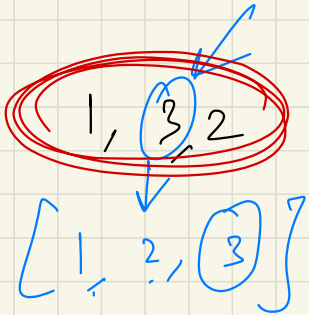
5, 4, 3, 2, 1

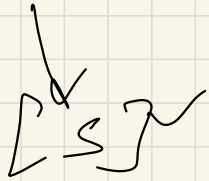
↳ Pivot: Choose any element → after first pass all the elements < p will be on LHS of p. & elements > p will be at RHS of p.

what?

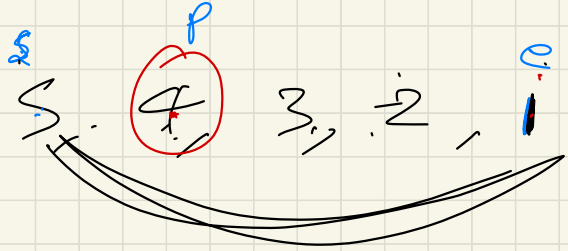
5, 4, 3, 2, 1



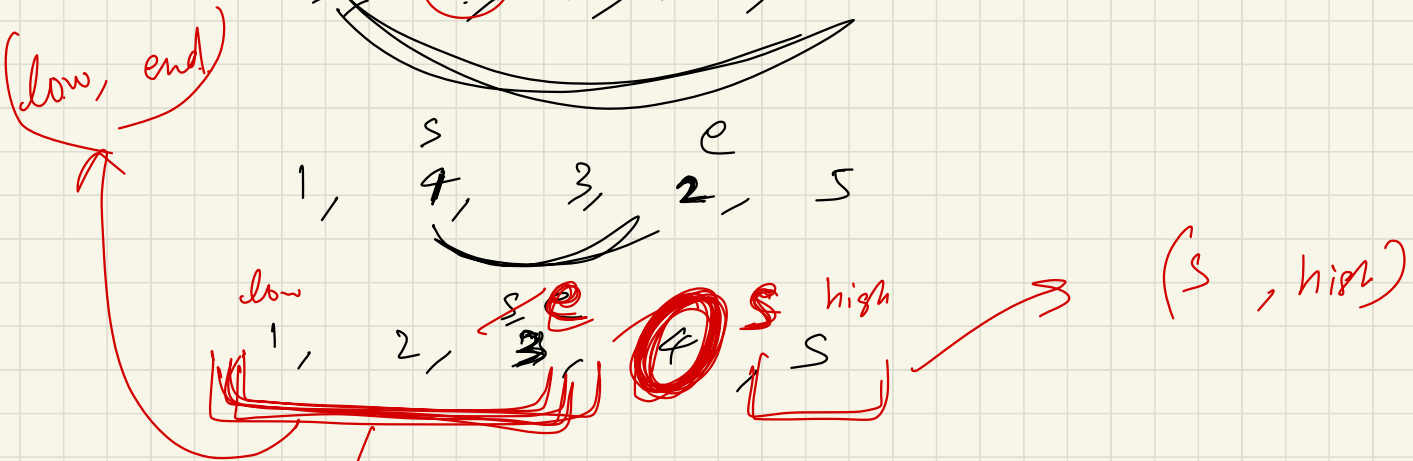


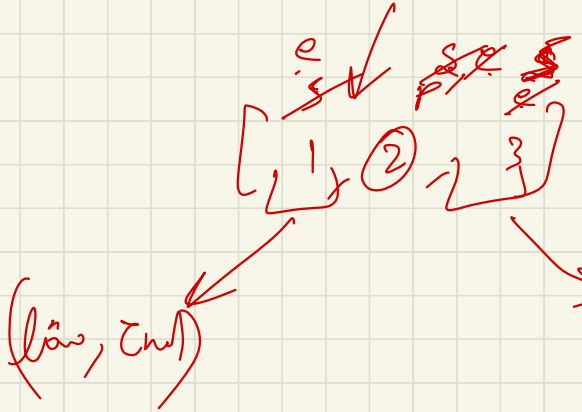


⊕ How to put pivot at correct position?



p = 4





while n[s] < p:
s++

while n[e] > p:
e--

S & end

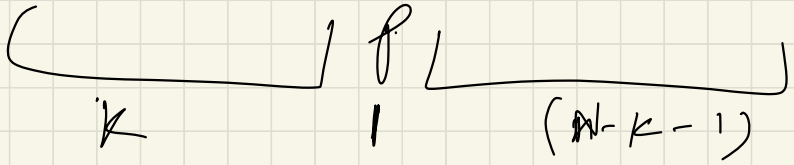
Swapping

low & high

main
tells us which part among
you are working on

How to pick pivot:

- Random element
- Corner element
- Pick the middle element



Important

$$T(N) = T(k) + T(N-k-1) + \Theta(N)$$

Worst Case:

$k = 0$

3, 5, 8, 9, 20, 34, 18, 66

$\downarrow$

$(n-1)$

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

$$\boxed{T(N) = T(N-1) + O(N)}$$

$$= O(N^2) \implies$$ Check SL & T complex lecture for complete derivation.

Best case:

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

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

$O(N \log N)$

Covered in detail the derivation of the exact formula.

- S & T courses
- MS Video

Notes:

- Not stable (already covered in initial sorting videos)
- In-place i.e why preferred for arrays instead of MS.
  MS takes $O(N)$ extra space
- MS is better in linked list due to memory allocation ⇒ not continuous

Hybrid sorting algorithms (Tim Sort):

Merge Sort + Insertion Sort

↓

Works well with partially sorted data