28 January 2025 12:11

Quick sort :-

Based on divide & conquer technique.

pivot | OP

<= pivot . 1st part Apply Algo

> pivot 2nd part Apply algo

pivot element ↓ आप Choose Karo . ↓ arr ka first/ last element

pivot = 35

35 50 15 25 80 20 90 45
P q

p → arr[p] <= pivot
q ← arr[q] > pivot

⊙ chk if p & q have crossed each other
↳ swap (pivot, arr[q])

⊙ chk if p & q havenot crossed each other
└ swap (arr[p], arr[q])

35 20 15 25 80 50 90 45
P q

n/2 - 1   n/2

25 20 15 | 35 | 80 50 90 45

OP

Left :- 25 20 15

pivot = 25
arr[p] <= pivot
arr[q] > pivot .

15 <= 25

swap (arr(q), pivot)

15 20 | 25 → OP

20 15 | 25 OP

80 50 90 45

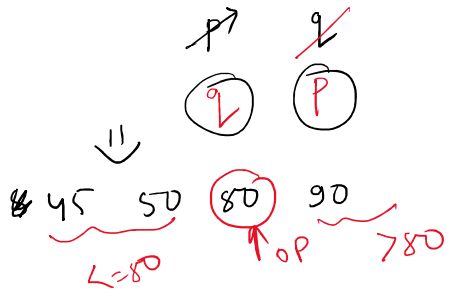
pivot = 80

arr(p) <= pivot .
arr(q) > pivot .

swap (p, q)

80 50 45 90

swap (pivot, q)



Merge Sort — RRelⁿ ⇒ $T(n) = 2T(n/2) + n$ — TC = O(n log n), SC = O(n)

Quick Sort — SC – O(1)

Time Complexity –

① 1

① Best Case

⊙ Pivot element → original position → middle pe

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

$= O(n \log n)$

② Worst Case – Sorted array as input

1 2 3 4 5

pivot = 1

1 | 2 3 4 5

n-1

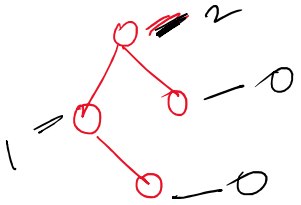
left sub | pivot

$T(n) = T(n-1) + n$

$= O(n^2)$

Trees –

Binary Trees – Nodes with 0, 1, 2 children.



Almost complete Binary Tree

→ Jab bhi nodes insert karo, toh left direction se karo.