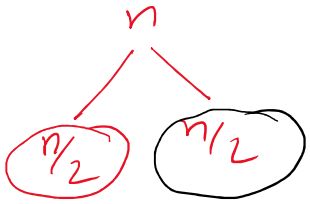
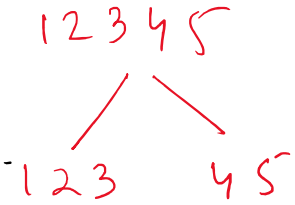
Merge Sort-



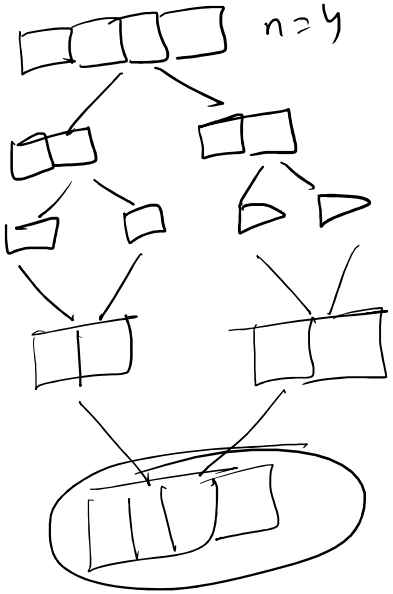
$T(n) = T(n/2) + T(n/2) + n$

$T(n) = 2T(n/2) + n \longrightarrow O(n \log n)$

SC → O(n)



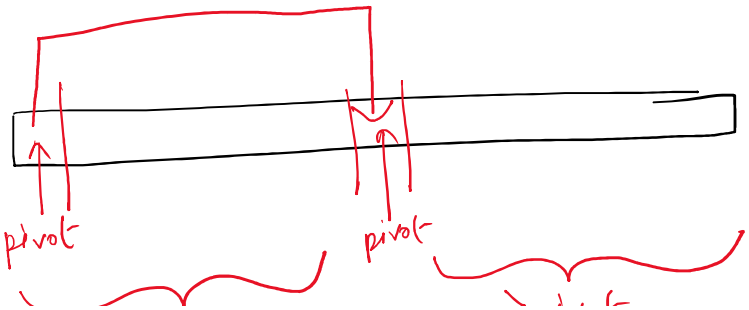
Quick Sort- Based on divide & conquer technique.



5 4 3 2 1

⇓

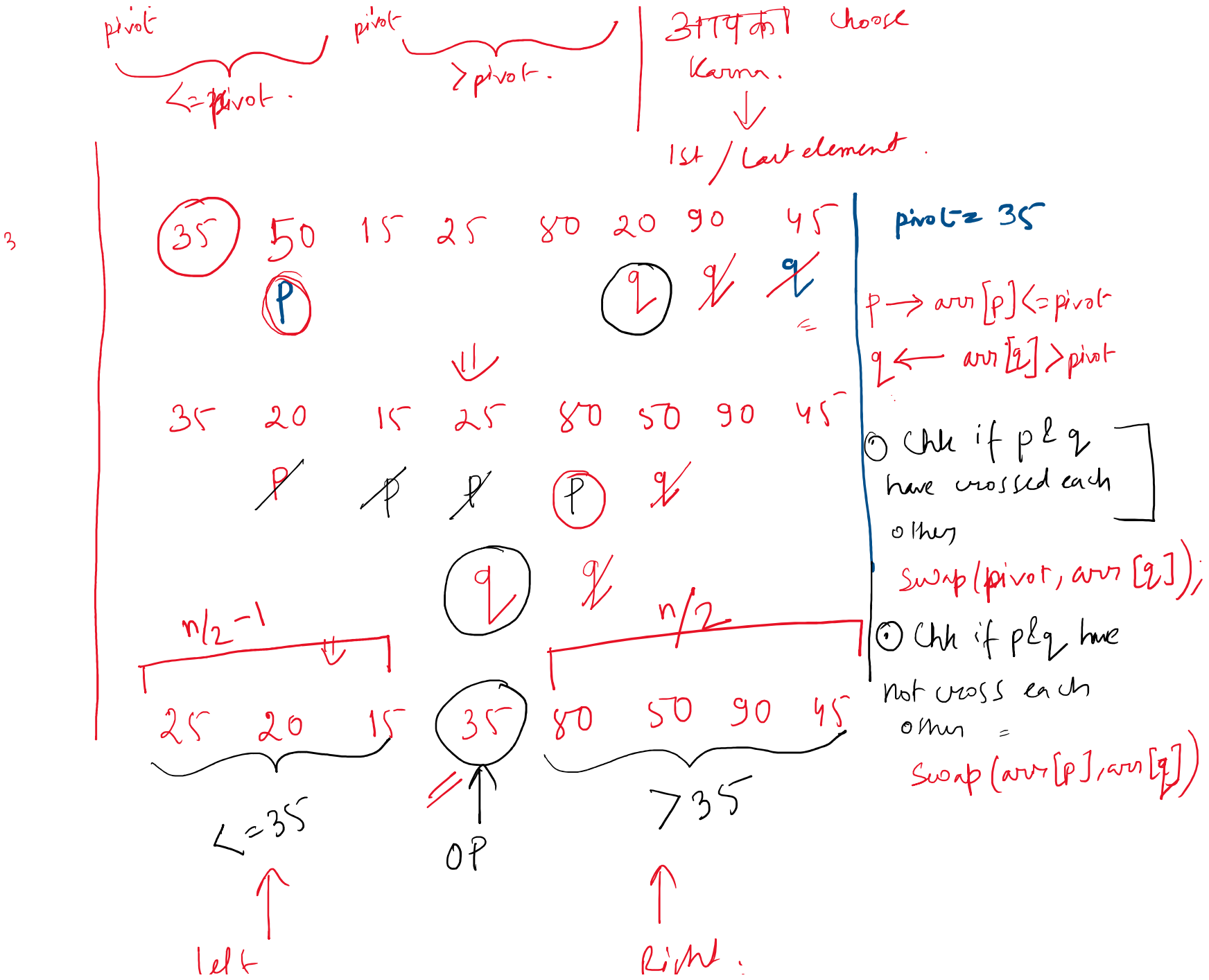
4 3 2 1 (5)



pivot element.

↓

choose


pivot
<=pivot.
pivot
>pivot.
choose
Karna.
1st / Last element.
3
35 50 15 25 80 20 90 45
P
q
pivot = 35
p → arr[p] <= pivot
q ← arr[q] > pivot
35 20 15 25 80 50 90 45
Chk if p & q have crossed each other
swap(pivot, arr[q]);
Chk if p & q have not cross each other
swap(arr[p], arr[q])
n/2 - 1
n/2
25 20 15
35
80 50 90 45
<=35
>35
OP
left
Right.

Left

Right

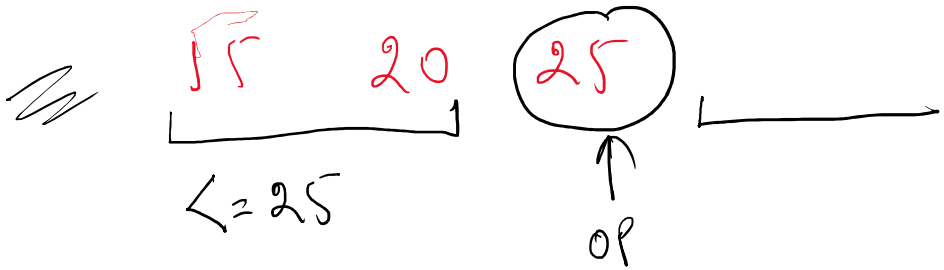
Left- 25 20 15

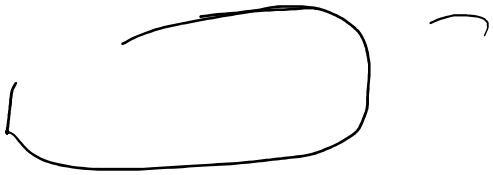
pivot = 25

arr(p) <= pivot

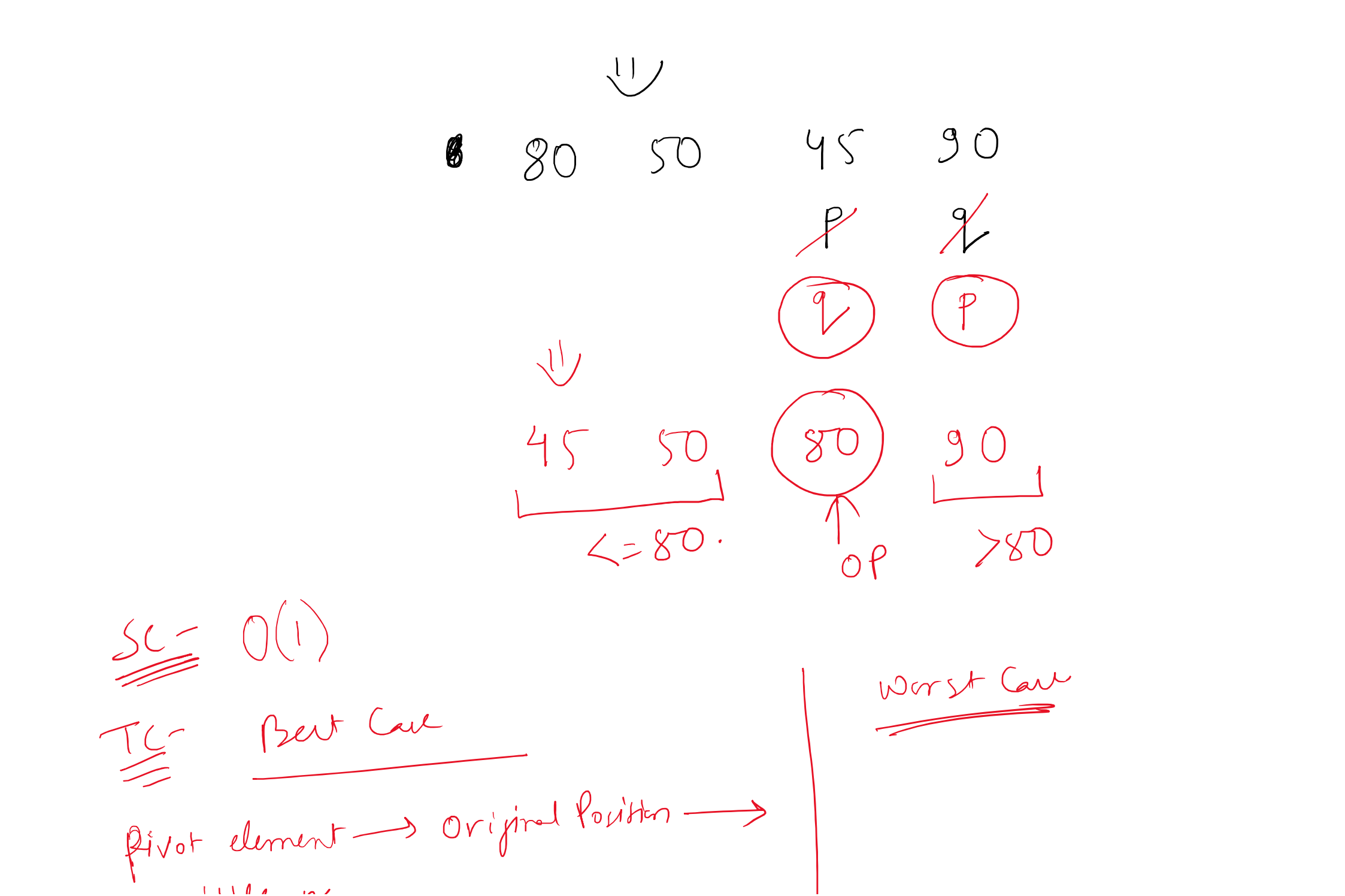
arr(q) > pivot

swap(pivot, arr(q))





Right 80 50 90 45 pivot=80



SC- O(1)

TC- Best Case

Pivot element → Original Position →

Worst Case

pivot element ... 0

middle pe.

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

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

$$TC = O(n \log n)$$