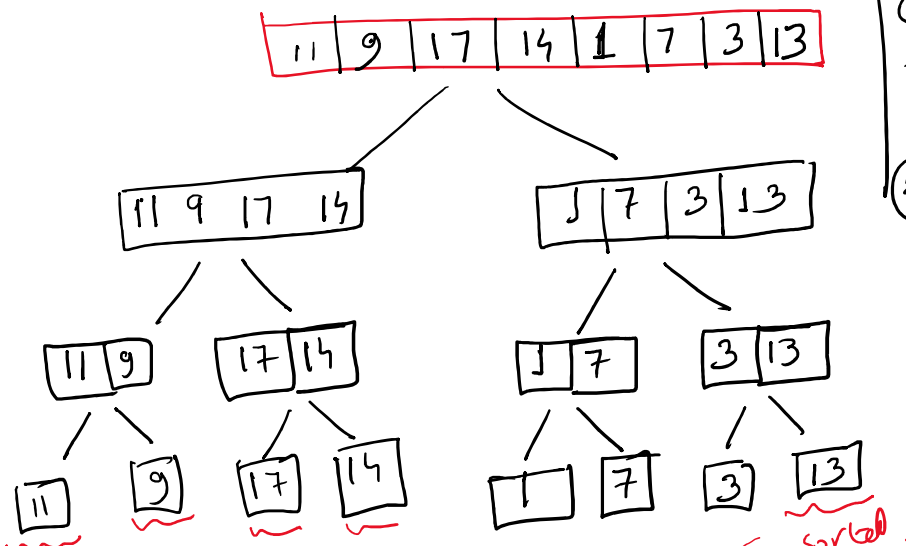
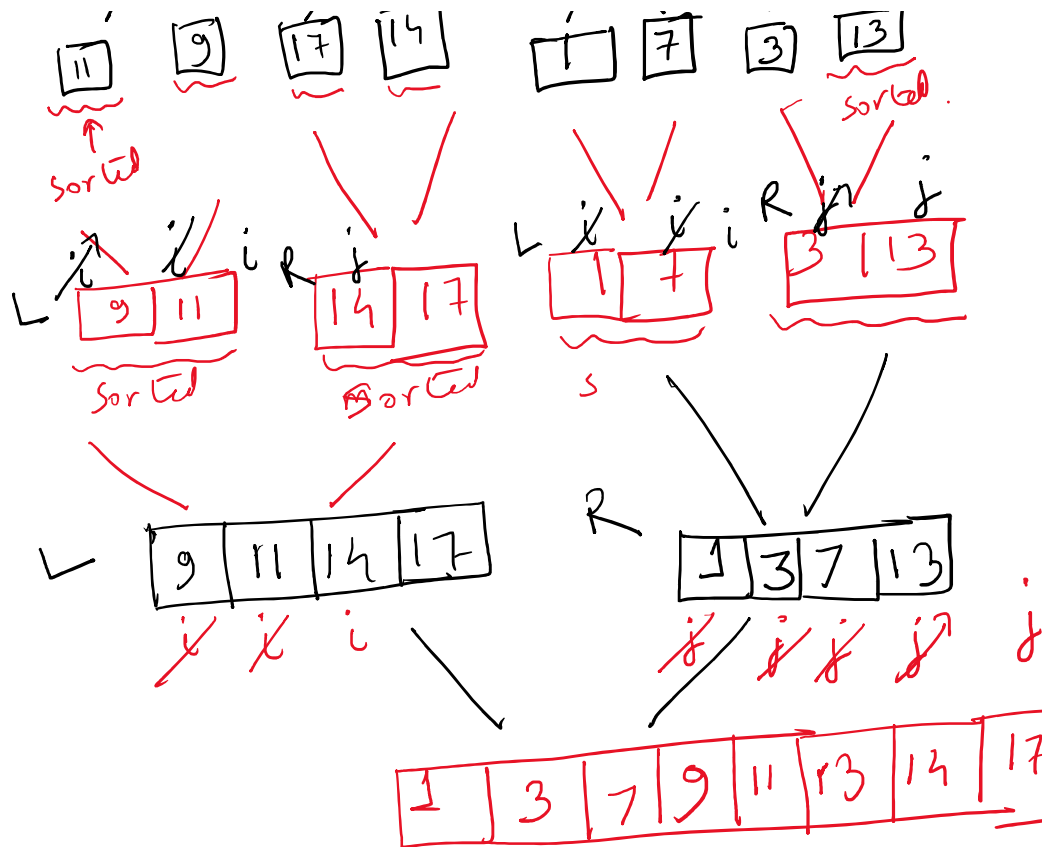


Merge Sort - Based on divide & conquer method.



Note -

- ① Repeatedly divide the array into 2 equal parts.
- ② Merge 2 sorted arrays into 1 sorted array.



Compare
 $L[i]$ $R[j]$

Time Complexity

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

Sub-problem
 solve same
 ka time

merge

n sized ka problem
 Time $\rightarrow T(n)$

$n/2$ ka size
 $T(n/2)$

① Repeatedly divide the array into 2 equal parts.

② Merge 2 sorted arrays into 1 sorted array.

na ...

|

$(n/2)$

arrays into 1 sorted array.

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

$$aT(n/b) \quad (n^k \log^p n)$$

$$a=2, b=2, k=1, p=0$$

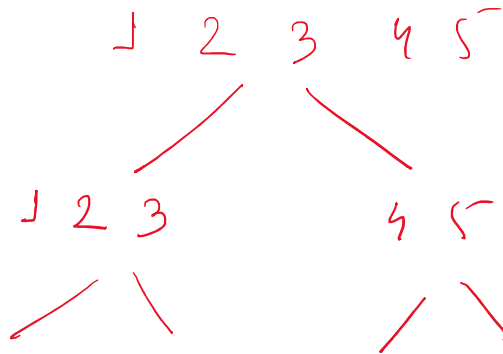
$$a < b^k$$

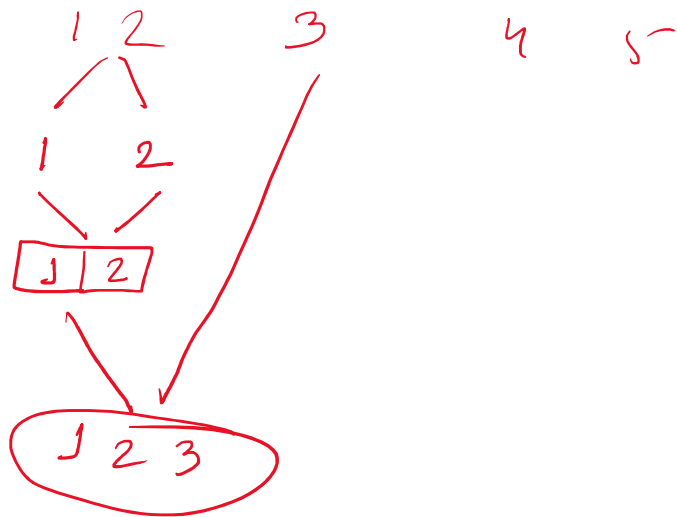
$$2 < 2^1$$

$$\text{Cond}^n 2a \rightarrow T(n) = O(n^{\log_b a, p+1}) = O(n^{\log_2 2, 0+1})$$

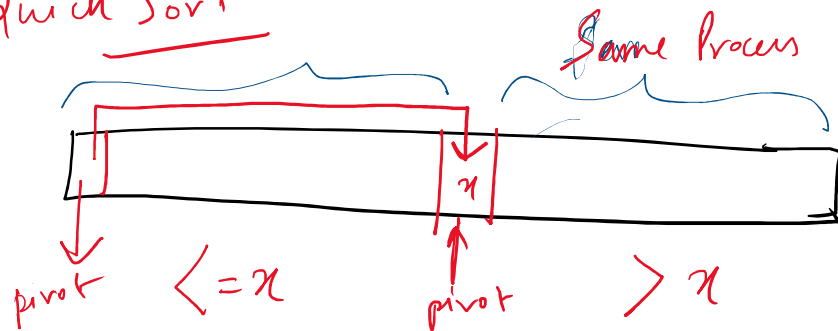
$$= O(n \log n)$$

$$SC = O(n) //$$





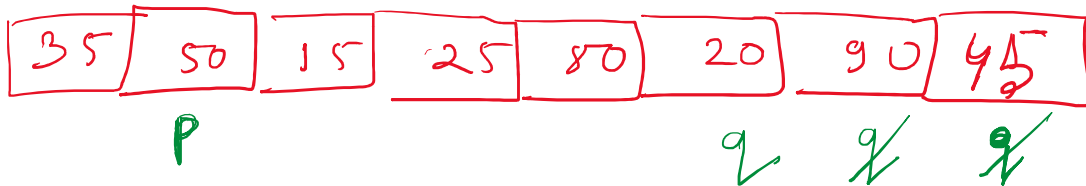
Quick Sort-



Note- pivot element

↓
Aapko choose
karna hain

pivot
= 35



35 20 15 25 80 50 90 45

Note-

① $p \rightarrow$

$arr[p] \leq pivot.$

② $q \leftarrow$

$arr[q] > pivot$

35 20 15 25 80 50 90 45

~~P~~ ~~P~~ ~~P~~ (P) ~~q~~
(2) ~~q~~

(25 20 15) (35) (80 50 90 45)
↑
Pivot

arr[q] > pivot

③ Chk if p and q have not crossed each other.

Swap(arr(p), arr(q))

④ If p and q crossed each other swap(arr(q), pivot);

Left array -

25 20 15

pivot = 25

~~P~~ → ~~q~~
~~P~~ P

15 20 25

Right p~-

80 50 90 45

~~P~~ →

2

P

pivot = 80

80 50 45 90

~~P~~

~~2~~

2

P

45 50 80 90