

Ans: we have given [14, 17, 13, 15, 19, 10, 3, 16, 9, 12]

• Apply Quicksort Algorithm:

(14) 17, 13, 15, 19, 10, 3, 16, 9, 12 + ∞
 Pivot element (we assume this is our Pivot Element) (P) Pointer
 // P Pointer search greater than 14 and stop when find greater value
 Q (Pointer) It search lesser than 14

\Rightarrow (14) 17, 13, 15, 19, 10, 3, 16, 9, 12 + ∞
 Pivot (P) $P > 14$ (yes)
 // then swap both values of P & Q
 (14) (12) (13) (15) 19, 10, 3, 16, 9, 17 + ∞
 Pivot P (13 > 14) (no) (15 > 14) (yes, stop) (Q) (9 < 14) (yes) // Now follow same process

\Rightarrow (14) 12, 13, 9, 19, 10, 3, 16, 15, 17 + ∞ again
 Pivot P (19 > 14) (yes) (Q) (3 < 14) (yes) (16 < 14) (no)

// Now again follow same process

\Rightarrow (14) 12, 13, 9, 3, 10, 19, 16, 15, 17 + ∞
 Pivot P (10 > 14) (no) (Q) (19 > 14) (yes)

\Rightarrow // when pointer P and Q cross each other then ... element with Q

do Partition means Replace the Pivot element with

Pointer element Like:

(10), 12, 13, 9, 3, (14), 19, 16, 15, 17, $+\infty$

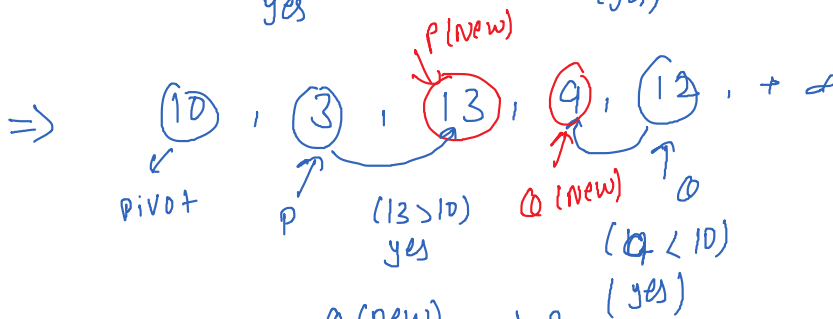
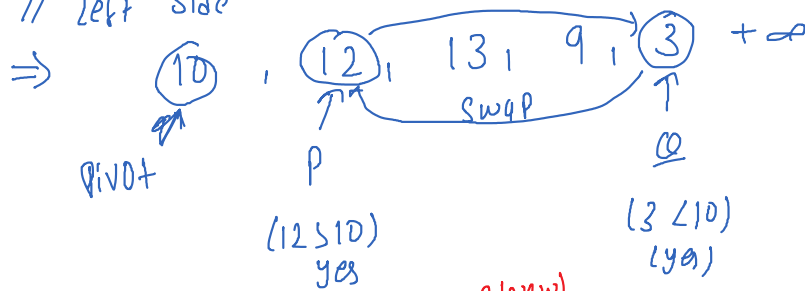
// look here this is our

first Partition where all smaller element from (14) is on left side and all greater values from (14) is on Right side.

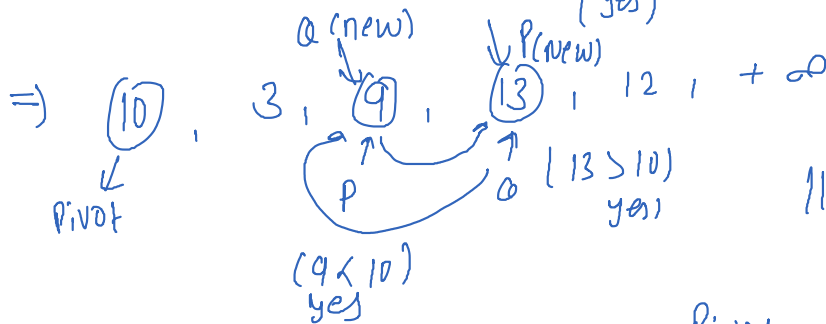
Left side (Apply Quick sort again)

Right side (Apply quick sort)

// Left Side



// swap again P & Q

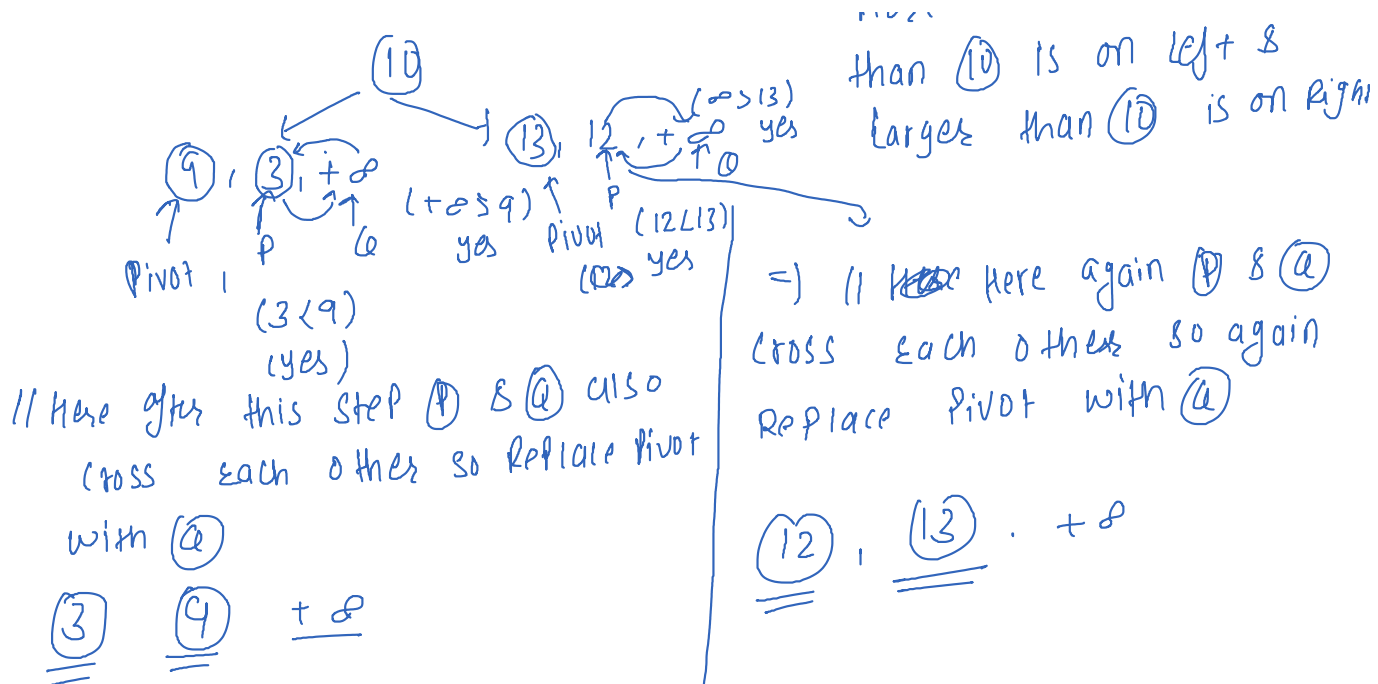


// now (P) & Q cross Each other so Replace Pivot element with Q

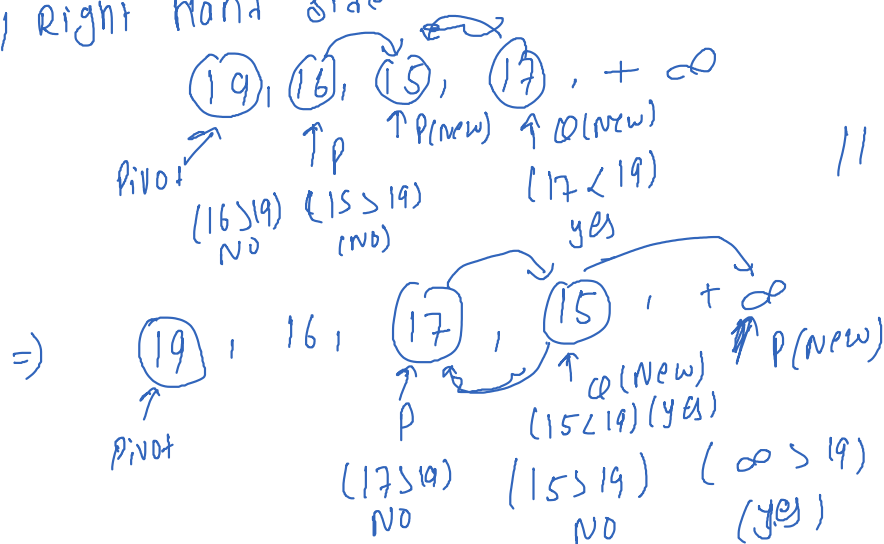
⇒ 9, 3, (10), 13, 12

— (10) — (10 > 13)

// Again we see a Partition here all elements smaller than (10) is on left & (10) is on Right



// Right Hand Side



=> 15, 16, 17, 19, + ∞ // Now we can see that the array of R.H.S is sorted

* Now take complete array (BOTH L.H.S & R.H.S) Together we see that our array is sorted

L.H.S R.H.S
3, 9, 12, 13, 14, 15, 16, 17, 19 // Now our array is sorted

L.H.S R.H.S
3, 9, 12, 13, 14, 15, 16, 17, 19 // now our array is
sorted