Page No.
Date / /

Time Complexity

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

left subtree right subtree

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

$$T(n) = 2^k T\left(\frac{n}{2^k}\right) + kn$$

$$2^k = n = n\,T(1) + \log n \cdot n$$

$$\log 2^k = \log n = n + n \log n$$

$$k \log_2 2 = \log n$$

$$k = \log n$$

Time complexity → $\boxed{T(n) = O(n \log n)}$

* Quick sort :-

28/11/25

7 , 6 , 10 , 5 , 9 , 2 , 1 , 15 , 7

start s swap end

→ pivot value (any value but)

left side | p | right side

≤ >

Quick Sort fails when values are already sorted

Page No.
Date

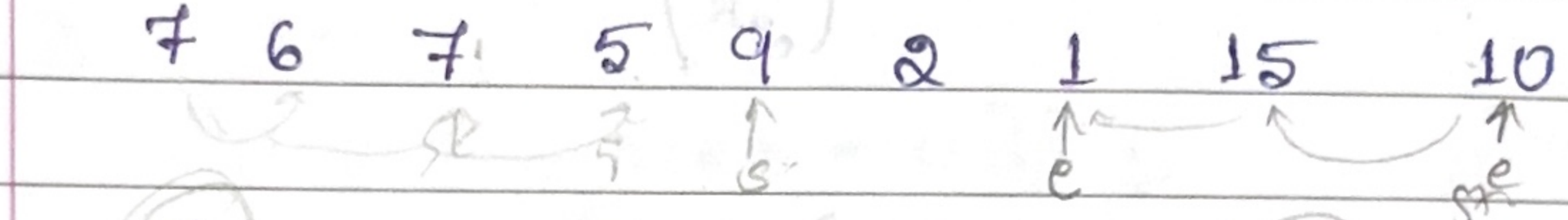
start ≥ end

when ⇒ value ≤ pivot
start ++
⟹ value > pivot
end --

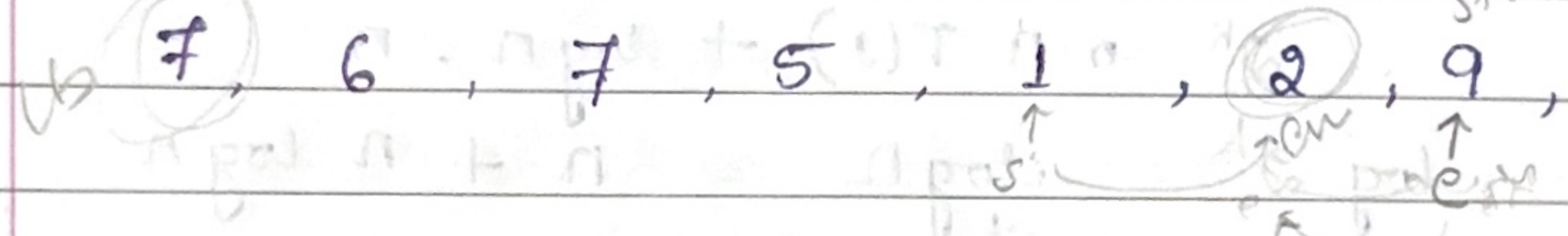
start & end cross each other
∴ swap lb with end

pivot value = 7

7 6 7 5 9 2 1 15 10

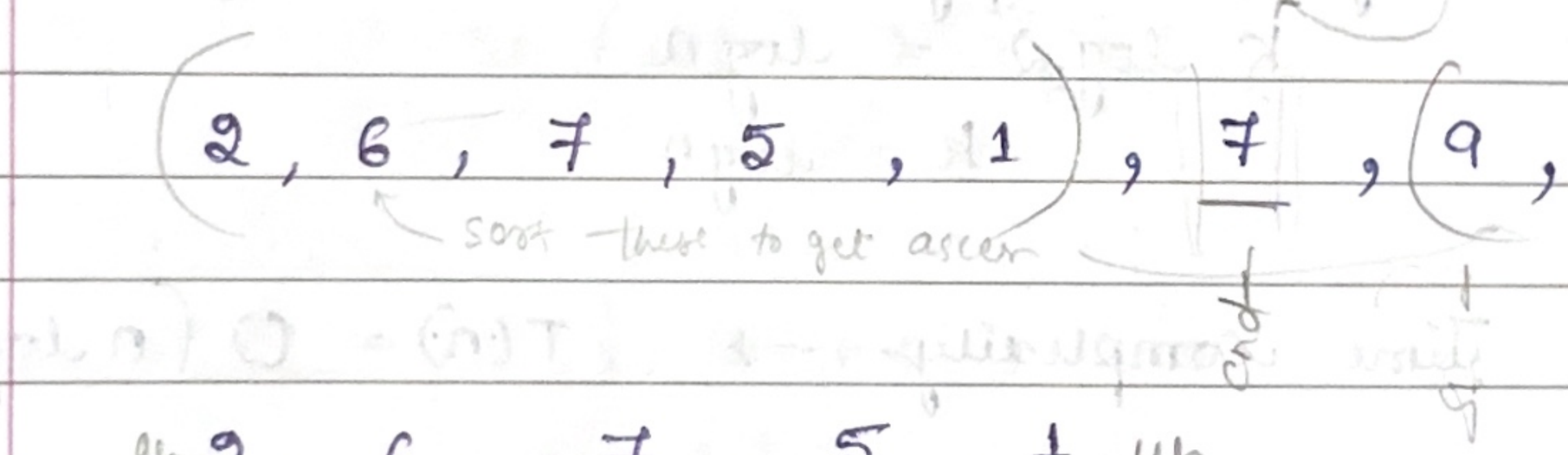


lb 7, 6, 7, 5, 1, 2, 9, 15, 10 ub



(2, 6, 7, 5, 1), 7, (9, 15, 10)

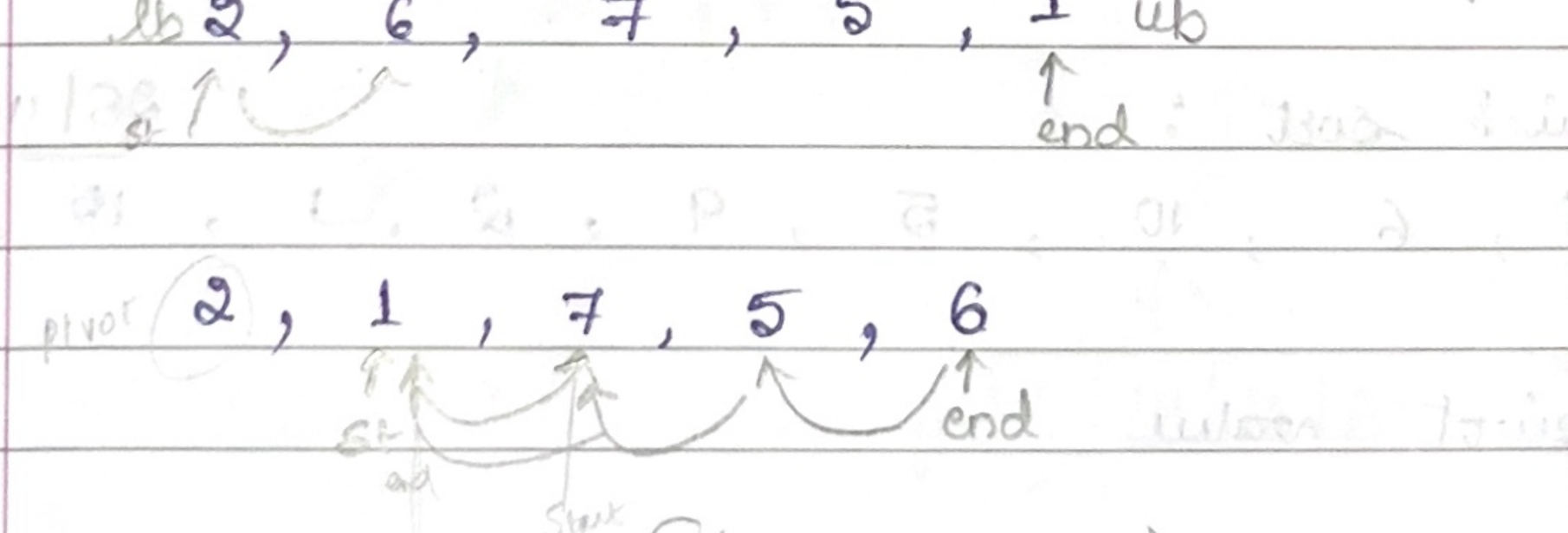
sort these to get ascen



lb 2, 6, 7, 5, 1 ub

s end

pivot 2, 1, 7, 5, 6



1, 2, (7, 5, 6)

sort

lb 7, 5, 6 ub

7 → pivot

start end

(6, 5), 7

lb ub
sort

end lb

5 , 6 , 7

pivot

Now lb 9 , 15 , 10 ub

swap 15, 10

9 , 10 , 15 .

∴ 1 , 2 , 5 , 6 , 7 , 7 , 9 , 10 , 15

```
Algo Quicksort (ar , lb , ub)
{
    if (lb < ub)
    pivot_idx = partition (A , lb , ub) ;
    Quicksort (A , lb , pivot_idx - 1) ;
    Quicksort (A , pivot_idx + 1 , ub) ;
}

partition ( A , lb , ub)
{
while (start < end)
{ ~~for~~ while (a[start] <= pivot)
        { start ++ ; }
while (a[end] > pivot]
        { end -- ; }
if (start < end)
{
```

```
        }
    }
    swap(a[lb], a[end]);
}
```

Time complexity → $\boxed{2T(n/2) + n.}$ → In best case

1 | 12 | 25

24 10, 20, 30, 40, 50

already in sorted order.

pivot value = 10

10, 20, 30, 40, 50

$(n-1) + (n-2) + (n-3) + (n-4) + \dots 0$

$n(n) - (1+2+3+4)$

$n^2 - \frac{n(n+1)}{2}$

$\boxed{TC = O(n^2)}$ → for worst case

* Heap Sort

1. Create Max heap
2. Delete root node & replace with element on right most from last level.