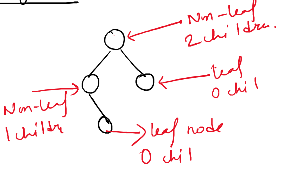
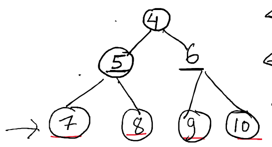
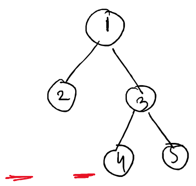
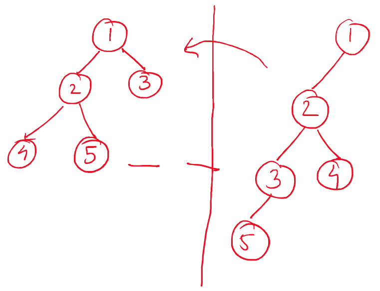
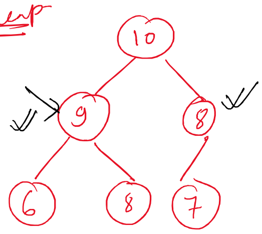
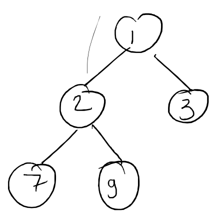
03 February 2025 12:56

Worst case TC of quick sort

Eg 1 2 3 4 → Input array $n = 4$

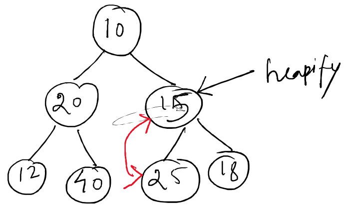
⇒ pivot = 1 , p = 2 , q = 4

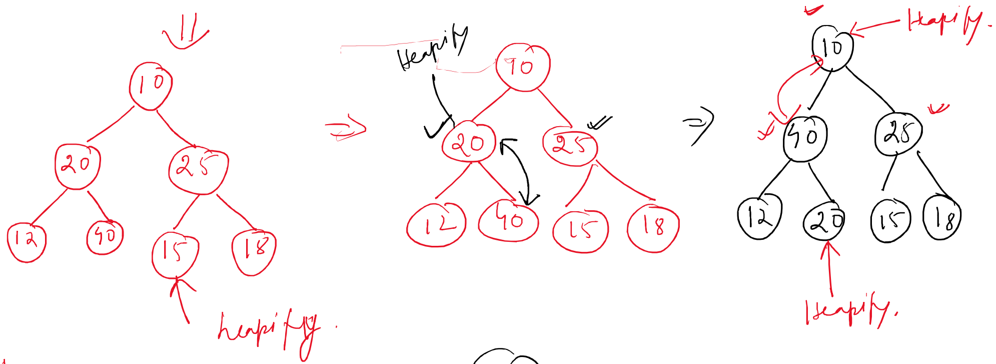
~~~ (1) 2 3 4

<= pivot OP >1

size of Right array = 3 = 4−1 = n−1

$T(n) = T(n-1) + n$ → substitution

$T(n) = O(n^2)$

Binary Tree – Tree → nodes → 0 / 1 / 2 childrens.



leaf node — Node → 0 children.

non-leaf node — Node → 1/2 children.

Almost complete Binary Tree – New nodes insert karo → Last level, → left direction

4 5 6 7 8 9 10



Max nodes in a level $= 2^L$



1 2 3 4 5.



2? ⇒

Heap –

1. ACBT
2. Max heap → Parents.val > Child.val

   Min heap → Parents.val < child.val.

Max heap



10 > 9 9 > 6

10 > 8 9 > 8

8 > 7.

Min heap



2? 1 2

HEAPIFY →
~~~

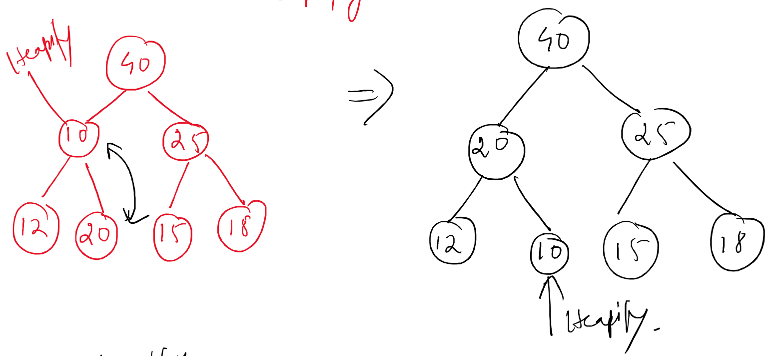
10 20 15 12 40 25 18. → Convert it into Max Heap.

→ Represent this array as ACBT.

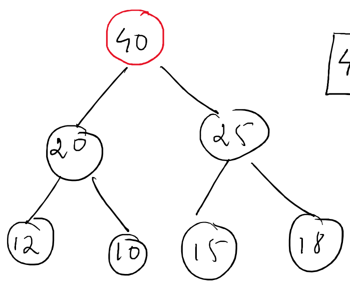
→ Last non-leaf node.







Array $\xrightarrow{\text{heapify}}$ Max Heap ———→ Sorted Order



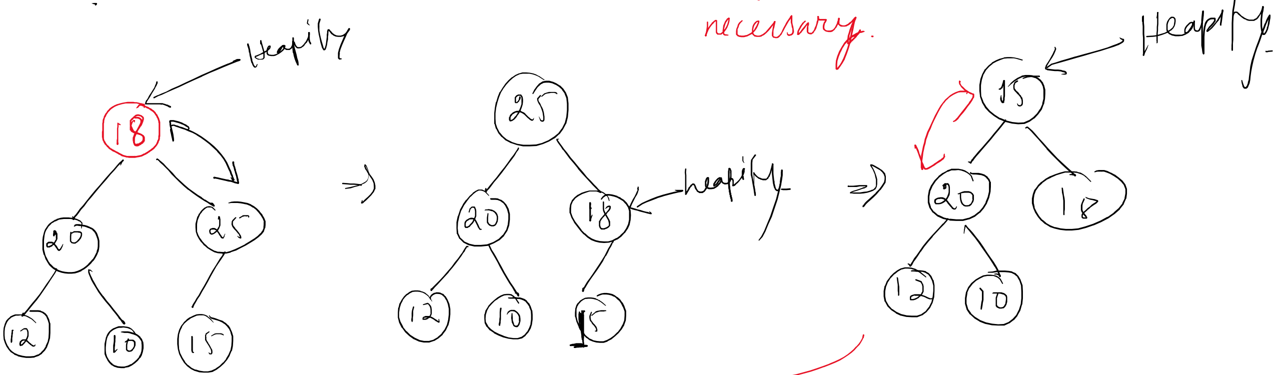
40	25

① Root value ko अलग से लिख लो

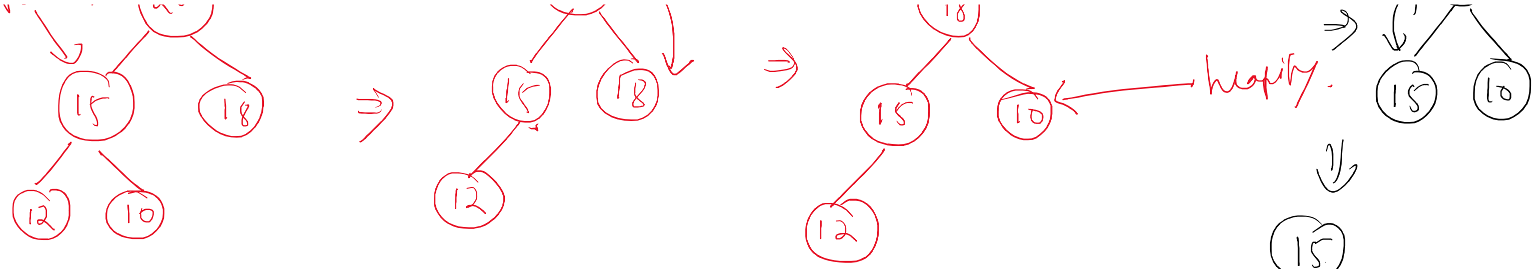
② Root→value = Last node→val

③ Delete last node.

④ Apply heapify whenever necessary.



40 25 20 18

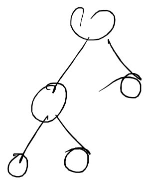


40 25 20 18 15 12 10

TC → O(

Height of ACBT = log n → n → no. of elements.

1 node insert → log n.



N nodes insert ⇒ n log n.

SC → O(1).