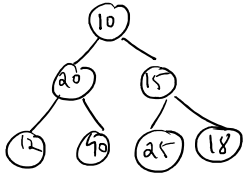


# Heap Sort -

10 20 15 12 40 25 18

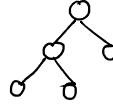


## Binary Tree

Tree

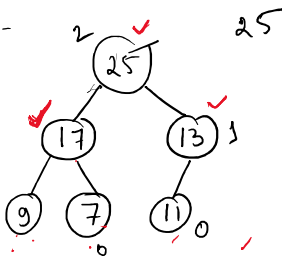
Nodes

0/1/2 children.



Max heap -  $\text{Node.value} > \text{child.value}$ .  
Min heap -  $\text{Node.value} < \text{child.value}$ .

Max heap -

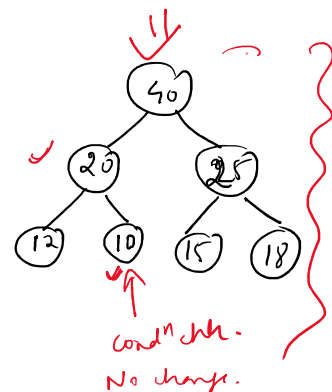
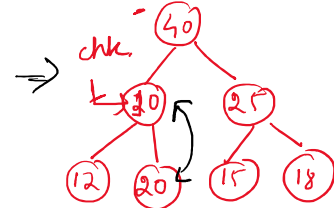
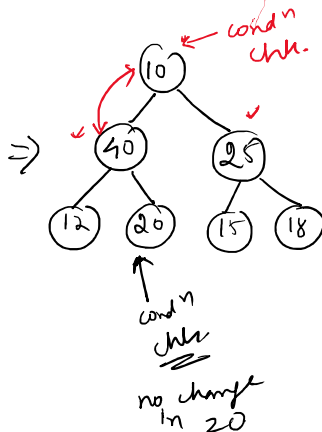
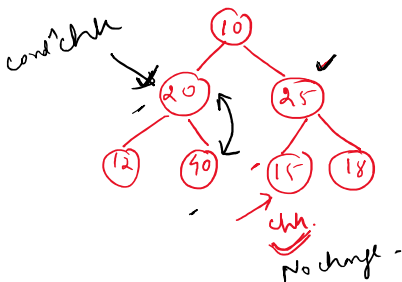
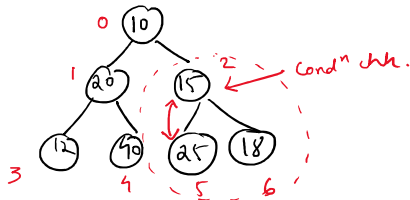


child	
17	25 > 17
13	25 > 13
leaf	non leaf
0 children	25 17 13
11 9 7	

Heapify - Array  $\xrightarrow[\text{apply}]{\text{heapify}}$  Max / Min Heap.

1/P arr - 10 20 15 12 40 25 18  
0 1 2 3 4 5 6

Convert it into tree.

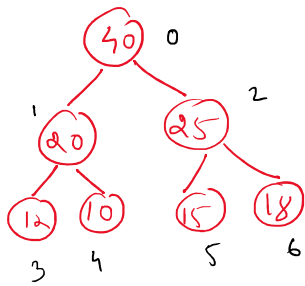


40 0

40 20 25 12 10 15 18

Algo

(1) Root-value, last node

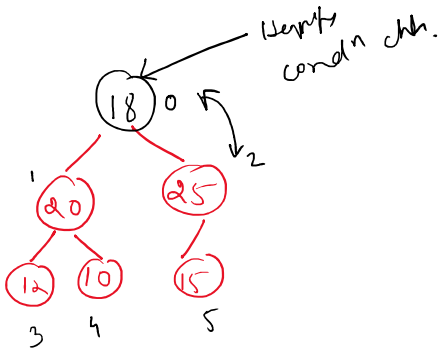


40 20 25 12 10 15 18  
0 1 2 3 4 5 6

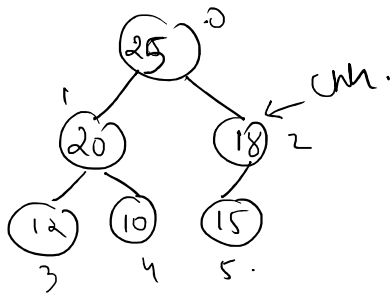
Algo

- ① Root-value, last node value swap.
- ② Last node delete.
- ③ Heapify (Root).

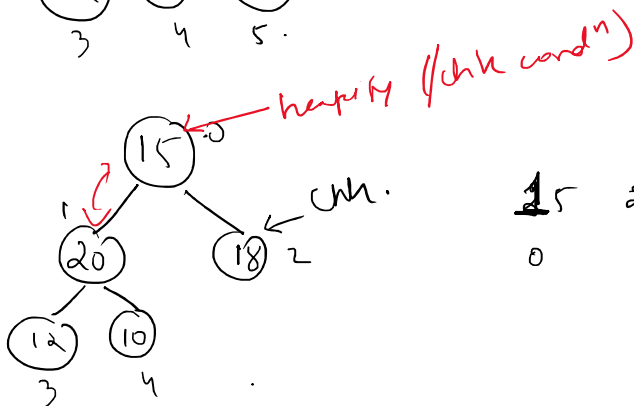
⇓



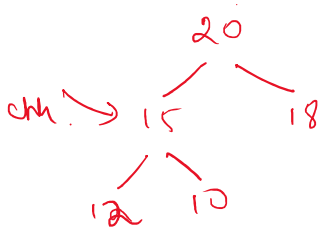
18 20 25 12 10 15 40  
0 1 2 3 4 5 6  
Sorted.



25 20 18 12 10 15 40  
0 1 2 3 4 5 6

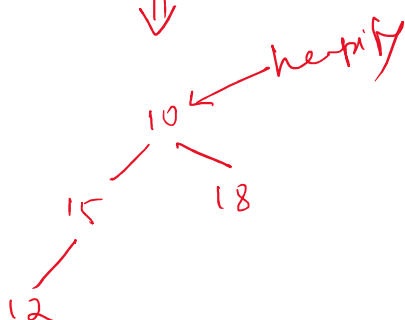


15 20 18 12 10 25 40  
0 1 2 3 4 5 6  
Sorted

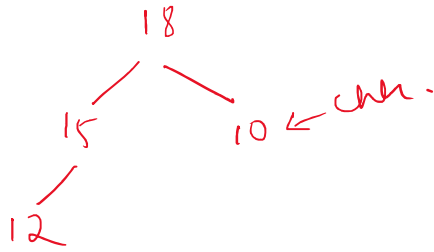


20 15 18 12 10 25 40

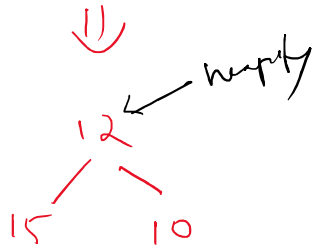
⇓



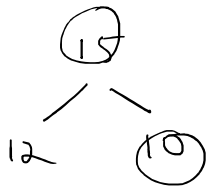
10 15 18 12 20 25 40  
Sorted.



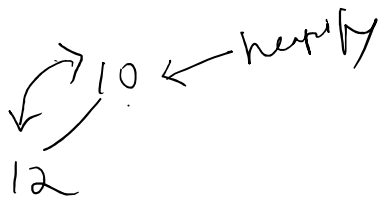
18 15 10 12 20 25 40



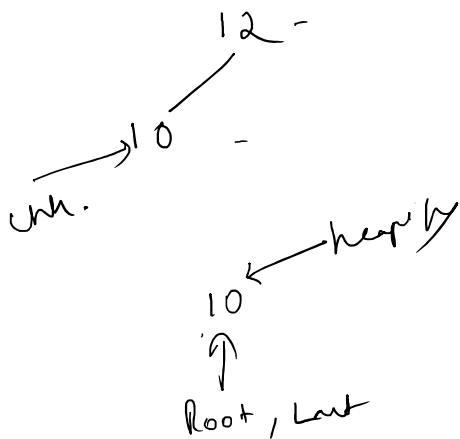
12 15 10 18 20 25 40  
sorted.



15 12 10 18 20 25 40  
sorted.



10 12 15 18 20 25 40  
sorted.



12 10 15 18 20 25 40

10 12 15 18 20 25 40  
sorted.

10 12 15 18 20 25 40  
sorted.