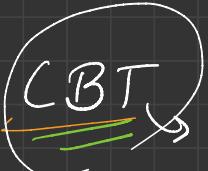



Heap

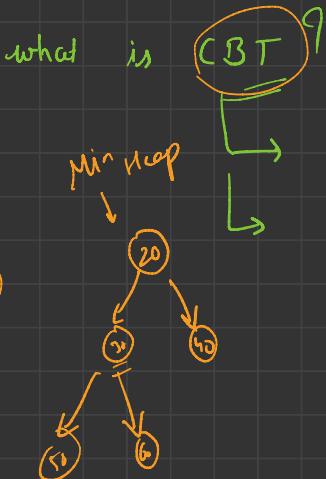
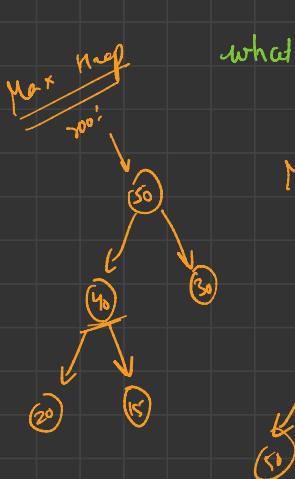
→ what is Heap?

is a

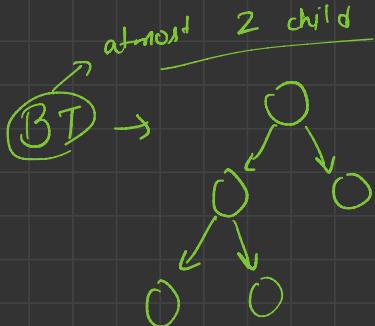


that comes with a
Heap Order Property

Max Heap Min Heap



every level is completely filled except the last level
nodes always added from the left
lean towards the left



→ insertion :-

Max Heap

$$\boxed{\text{val} = 55}$$

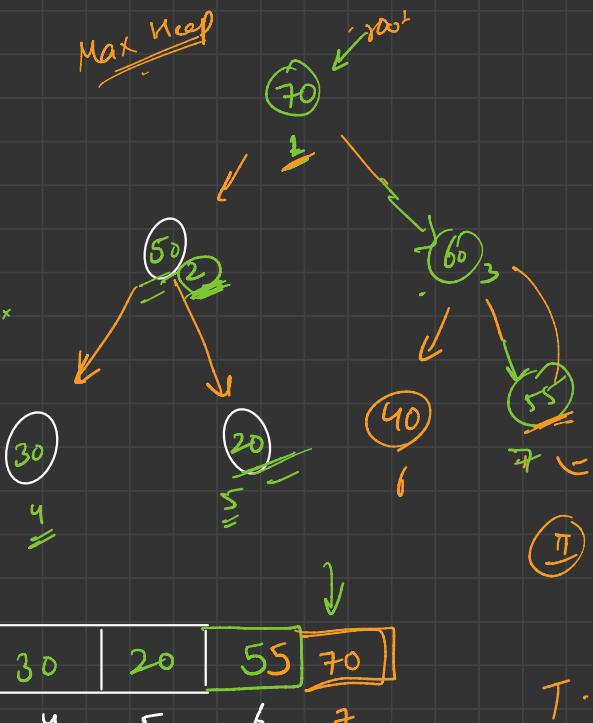
$$\boxed{\text{val} = 70}$$

Node → i^{th} index

left child → $2 \times i$ index

right child → $(2 \times i + 1)$ index

parent = $(i/2)$



X	60	50	40	30	20	55	70
0	1	2	3	4	5	6	7

→ inserted end

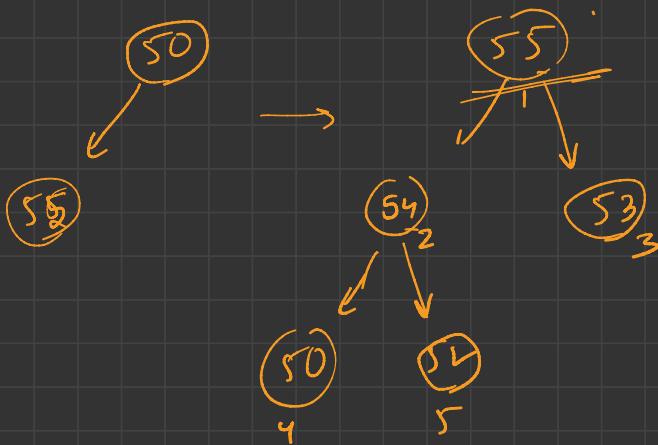
II → take it to its correct pos

$T.C \rightarrow O(\log n)$

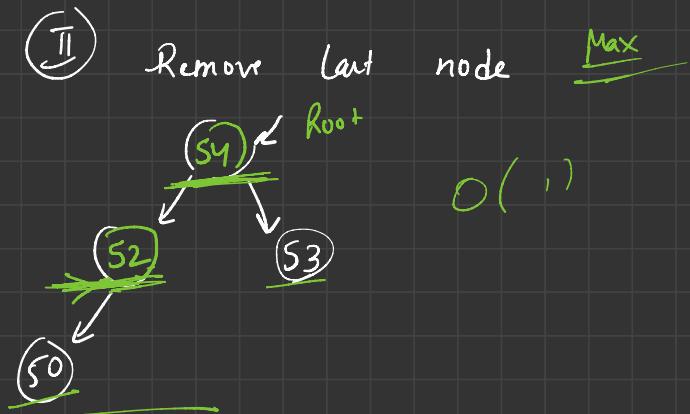
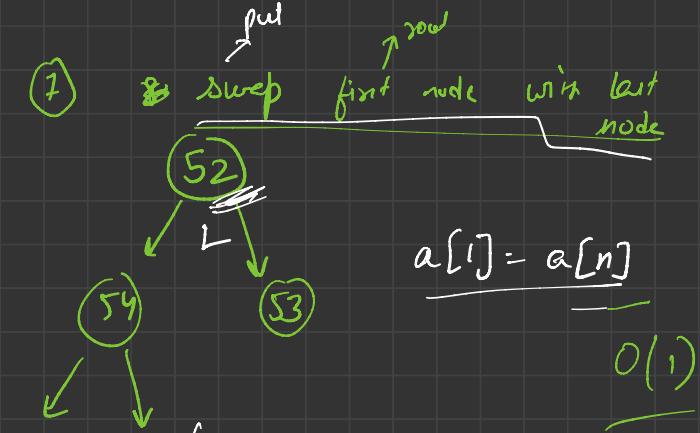
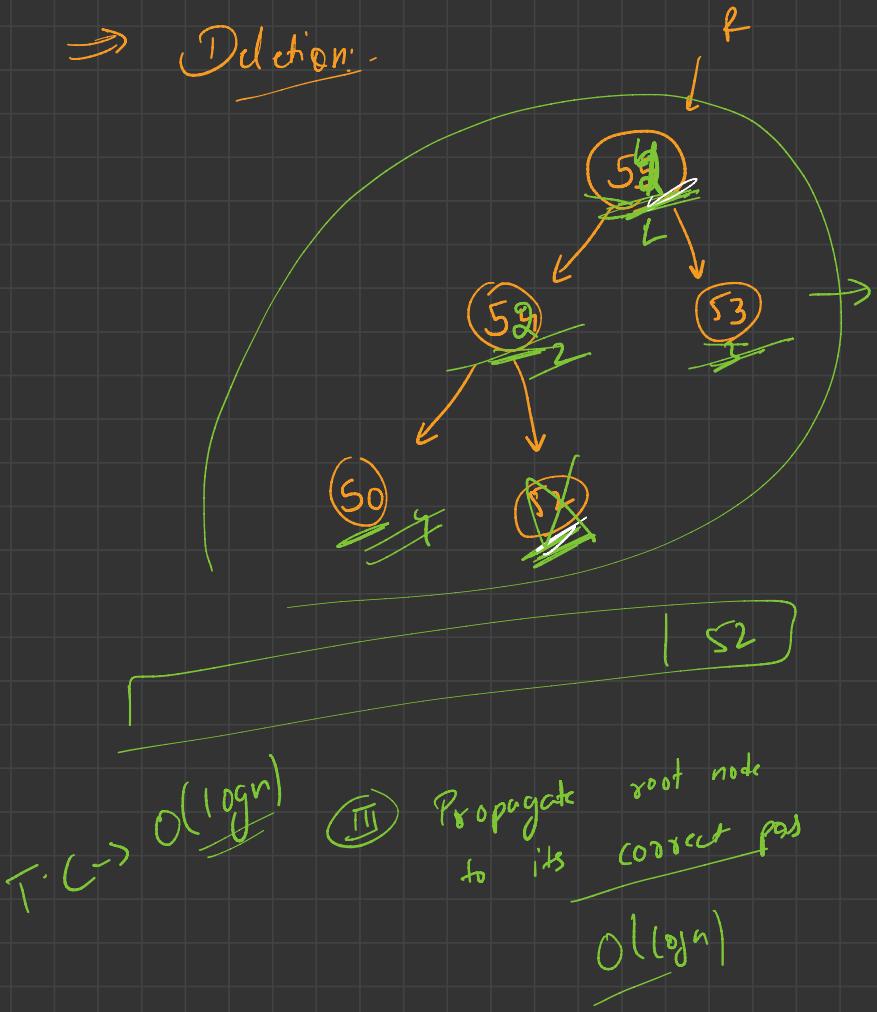
→

50, 55, 53, 52, 54

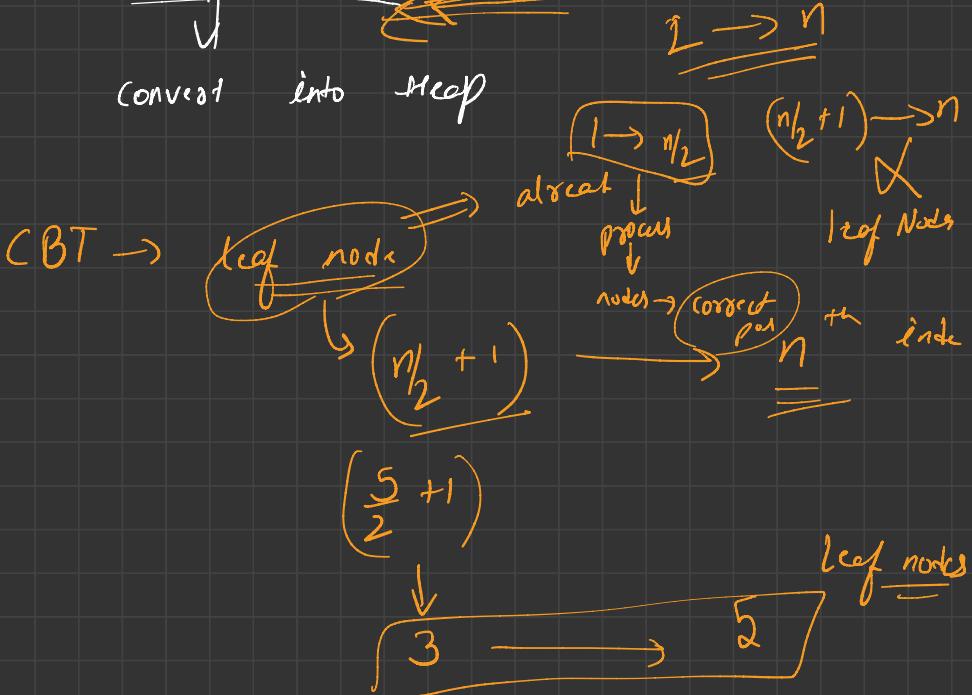
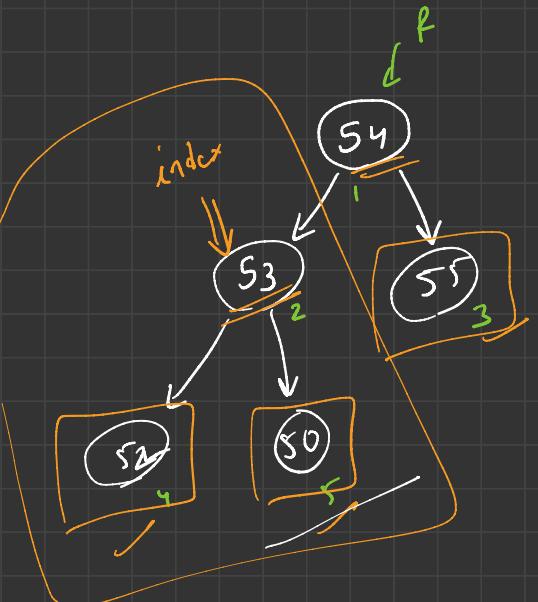
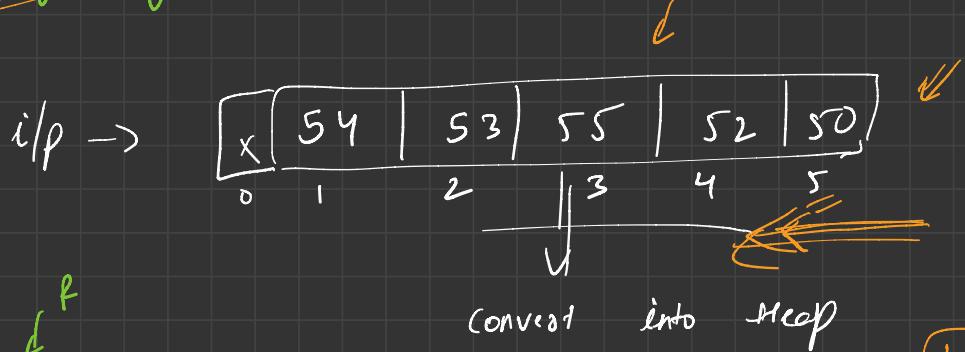
root



\Rightarrow Deletion:



\Rightarrow Heapify algo $\Rightarrow O(\log n)$



for ($\frac{i=}{n/2} \rightarrow >0$)

{

 heapify (arr, size, i);

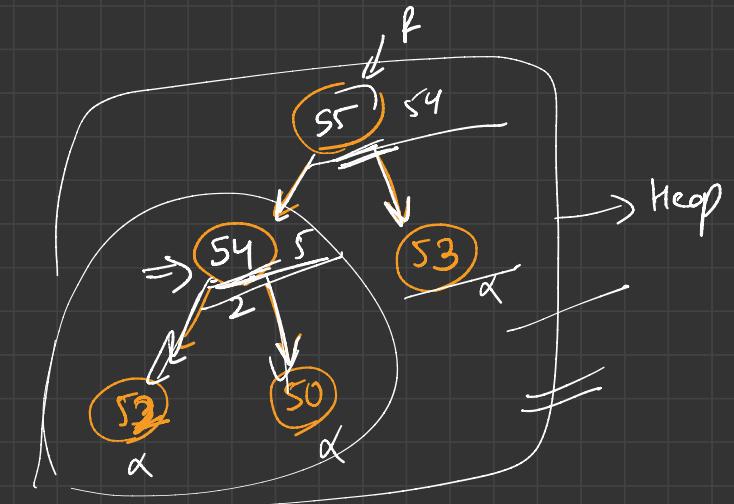
}

buildHeap

O(n)

How

4/ω

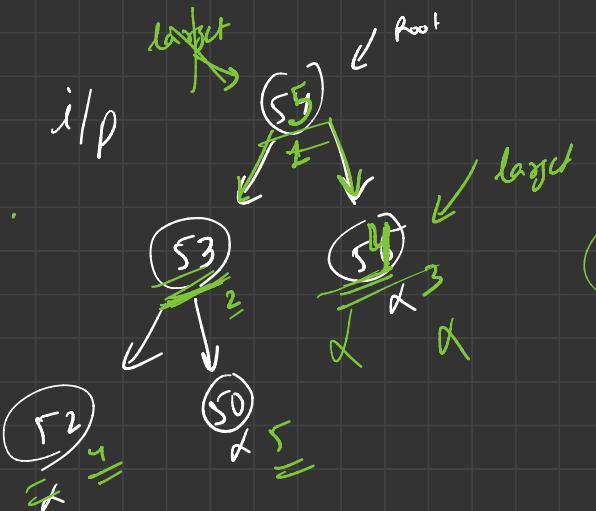


i bound

$$\begin{aligned}left &= 2^i \\right &= 2^i + 1\end{aligned}$$

$i = 0, 0$ bound is

$$\begin{aligned}left &= 2^i + 1 \\right &= 2^i + 2\end{aligned}$$



55 53 54 52 50



insertion

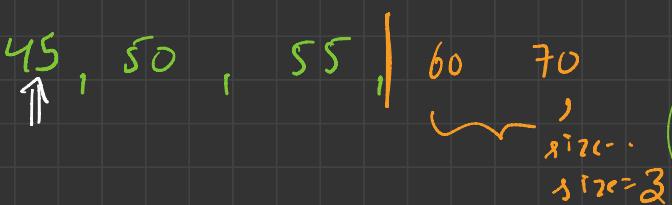
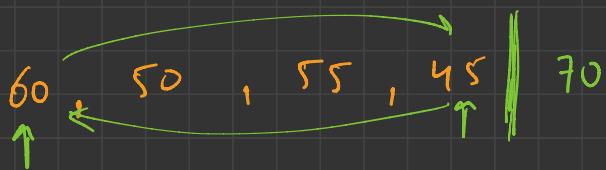
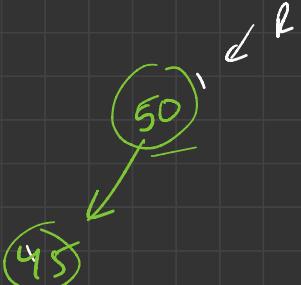
deletions

heaps

BuildHeap

Heap sort $\rightarrow O(n \log n)$

\Rightarrow Max-Heap Soot:



Algo:-

while(heap[1])

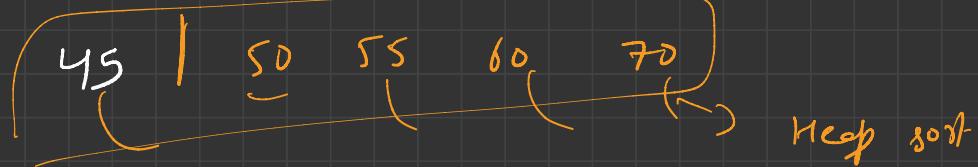
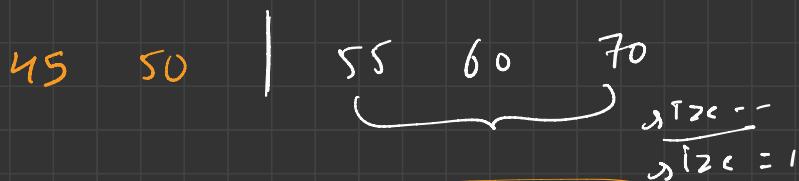
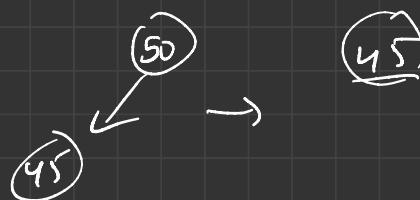
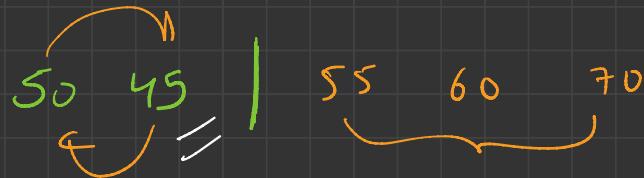
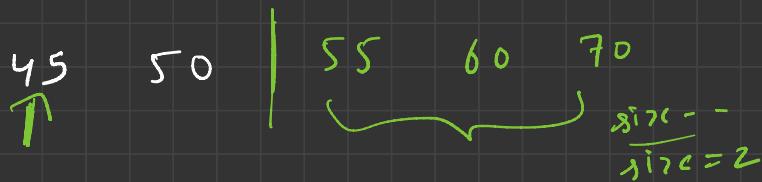
 root

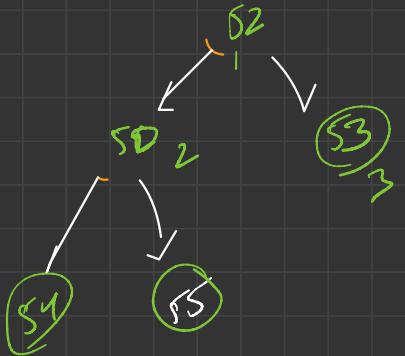
 last node

 (1) swap a[1] -> a[n]

(II)

 root node \rightarrow correct pos
 \downarrow
heapsify





$O(n \log n)$

arr \rightarrow ip

heapsort ()

↓

// \rightarrow BuildHeap

// Heapsort logic we would hear

ζ

