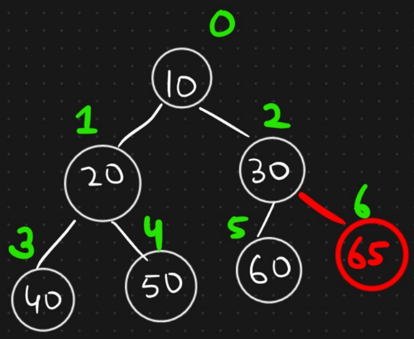
Deletion in Minheap

Insertion

- worst & average $= O(\log n) \longleftrightarrow$ n elements
- Best case $= O(1) =$ constant



$30 < 65 \Rightarrow$ valid Minheap

1 comparison &

0 swap

$$\# \text{elements} = n * 2^n \text{ (Insertion)} \quad n * \cancel{\log_2 2} \downarrow 1$$

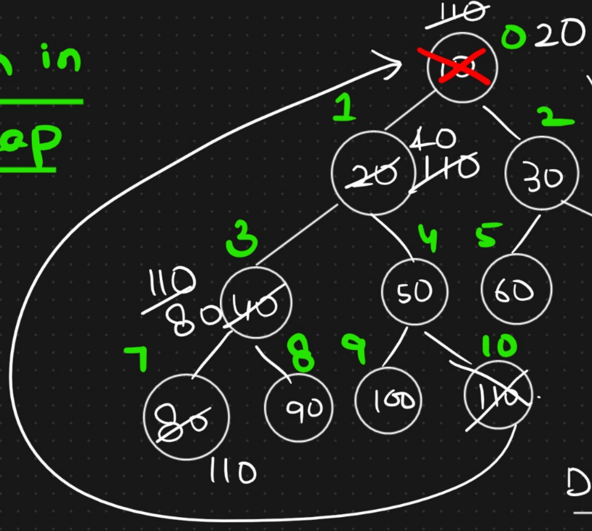
$$= \log_2 (n * 2^n)$$

$$= \log_2 n + \boxed{\log_2 * 2^n}$$

$$= \log_2 n + n$$

$$= O(n) \Longleftarrow$$

Deletion in minheap



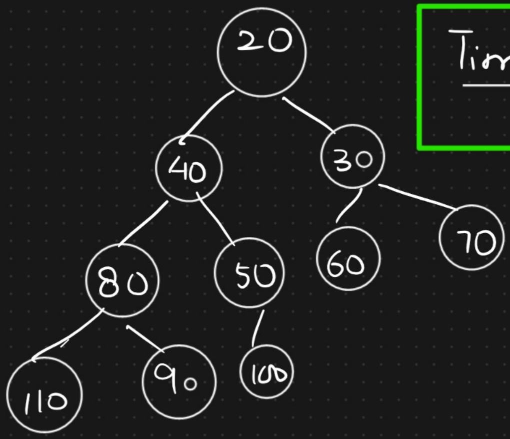
Valid Minheap { 1) CBT
2) Parent < child node
data data

⇒ 10
Delete

Deletion → # comparisons
↳ 2 * logN
⇓
Left child
&
Right child

Root Node to
Leaf Node }

swaps → logN



Time complexity → O(logN)

comparison based sorting
↑
Heapsort

Deletion - 1st time → 1st smallest (logN)

——— 2nd time → 2nd smallest (logN)

——— 3rd time → 3rd smallest (logN)

⋮

——— n time → nth smallest element (logN)

Array
⇓
Sorted array
⇓
increasing order

Overall time complexity = O(nlogn)

Maxheap → sorted array

Deletion

n number

of times

↳ Decreasing order

$\boxed{\underline{\text{Heapsort}} \rightarrow \underline{O(n \log n)}}$