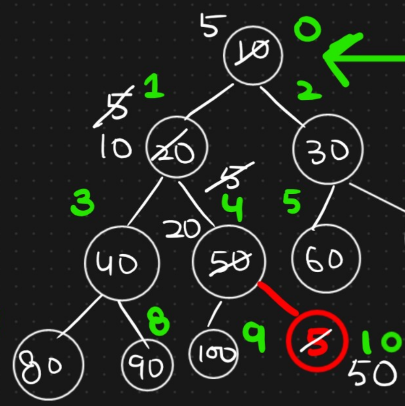
Insertion in Minheap / Maxheap

$n = 10$

Complete Binary Tree



$x = 5$

↳ New data element

$50 < 5 \rightarrow$ Not true

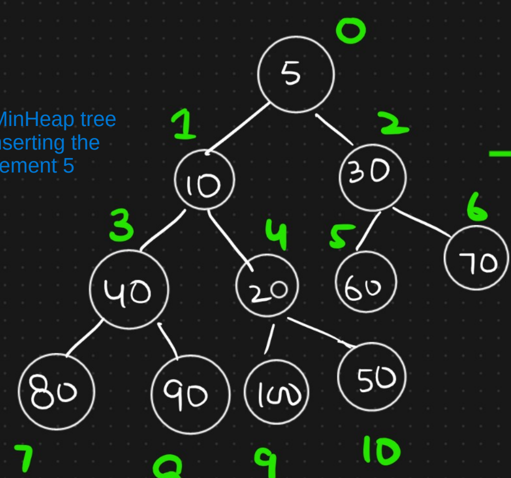
$50 > 5$
$20 > 5$
$10 > 5$
$\} \rightarrow$ 3 swaps

0	1	2	3	4	5	6	7	8	9	10
10	20	30	40	50	60	70	80	90	100	5

Initially new data point 5 is stored at the next available index that is 10. But since it's violating the property of MinHeap that is parent node < Child node so we will perform continuous swapping until new data point satisfies or is in accordance with property as defined by MinHeap

$\log N$

Valid MinHeap tree after inserting the new element 5



→ Valid minheap Tree

comparisons = $\log N$

swaps = $\log N$

In worst case we need to move the new inserted element to the root node from leaf level. Hence no.of comparisons and number of swaps will be equivalent to number of levels (since moving new node data from leaf level to root level) which is K = log(N)

0	1	2	3	4	5	6	7	8	9	10
5	10	30	40	20	60	70	80	90	100	50

$n = 2^k - 1 \rightarrow$ complete binary tree

$(n+1) = 2^k$

$k = O(\log n)$

$\log_2(n+1) = \log_2 2^k$

$$\boxed{\log_2(n+1) = k \cancel{\log_2 2}}$$

Time complexity = Number of comparisons + Number of swaps

- Insertion (Minheap/Maxheap)

In worst case for inserting an element in the MinHeap, time complexity will be of order of log N
O(log N)

$$\boxed{= O(\log N)}$$