Data Science & Artificial Intelligence

Data Structures Through Python

TREES

Lecture No.- 04

By- Satya sir

Recap of Previous Lecture

- Tree Traversals
  - Breadth-First : Level-order
  - Depth-First : In order, Pre order, Post order
    (LPR) (PLR) (LRP)

- Binary search Tree : Left subtree < Parent < Right subtree
  - Construction
  - Insertion

- Inorder Traversal of a BST will Output Elements in Ascending order.

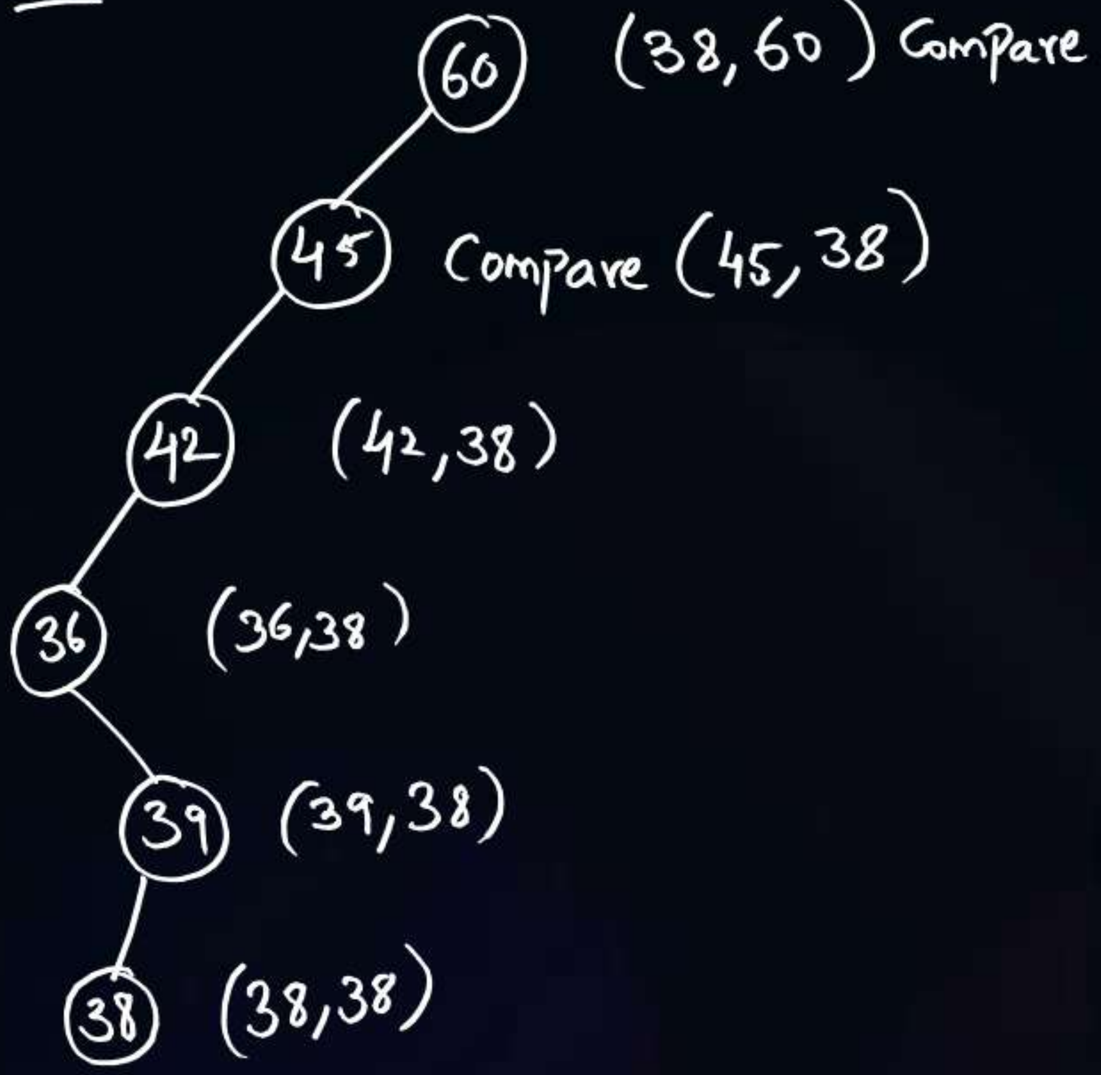
Topics to be Covered

- Search operation in a BST
- Deletion operation in a BST
- Binary Heap
  - Definition
  - Insertion operation

Topic : Binary Search Tree

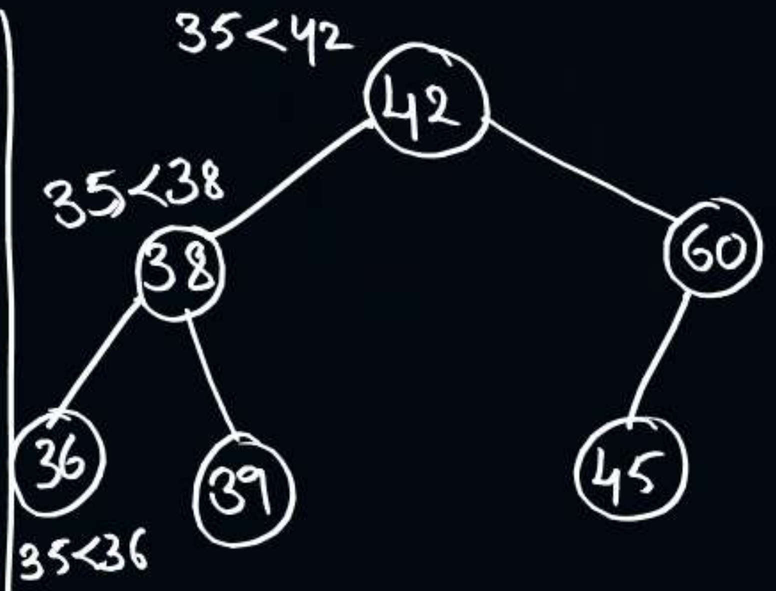
Search for an Element 38 / 35 == key

Ex:



Worst Case : key is found at last Comparison
(or) Not found
$O(n)$

42, 60, 45, 38, 36, 39



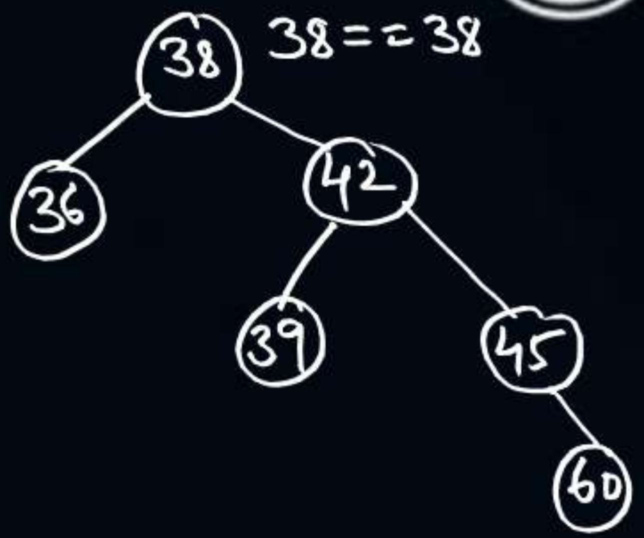
Key = 35

No of comparisions = $\lceil \log_2^n \rceil - 1$

Time Complexity = $O(\log_2^n)$

Average case

38, 42, 36, 39, 45, 60



Key = 38

No of Comparisons == 1

Best Case : $O(1)$

Topic : Binary Search Tree

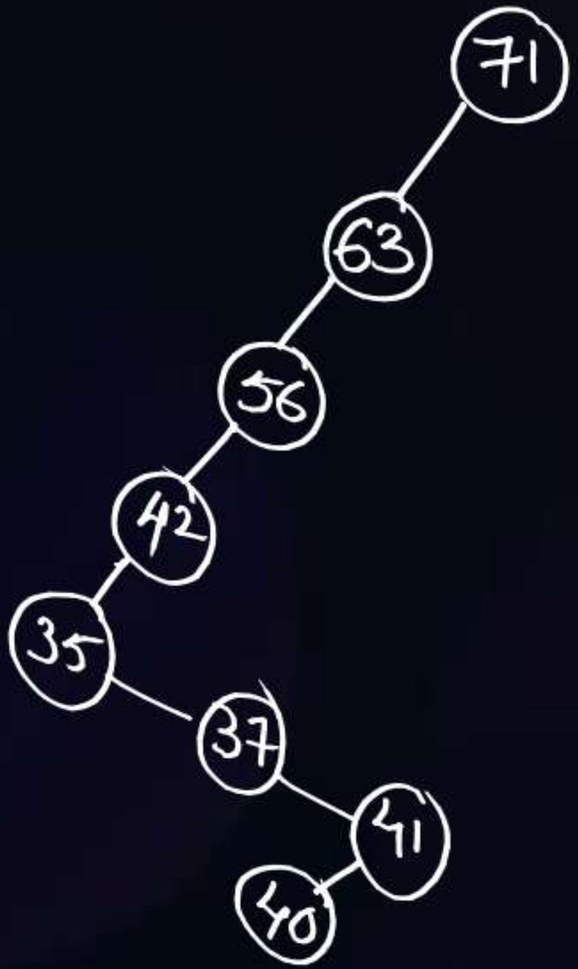
H/W Question

BST

Post-order : 40, 41, ~~37~~, ~~35~~, ~~42~~, ~~56~~, ~~63~~, ~~71~~

In-order : ~~35~~, ~~37~~, 40, ~~41~~, ~~42~~, ~~56~~, ~~63~~, 71

BST



Level order = 71, 63, 56, 42, 35, 37, 41, 40

Pre order = 71, 63, 56, 42, 35, 37, 41, 40

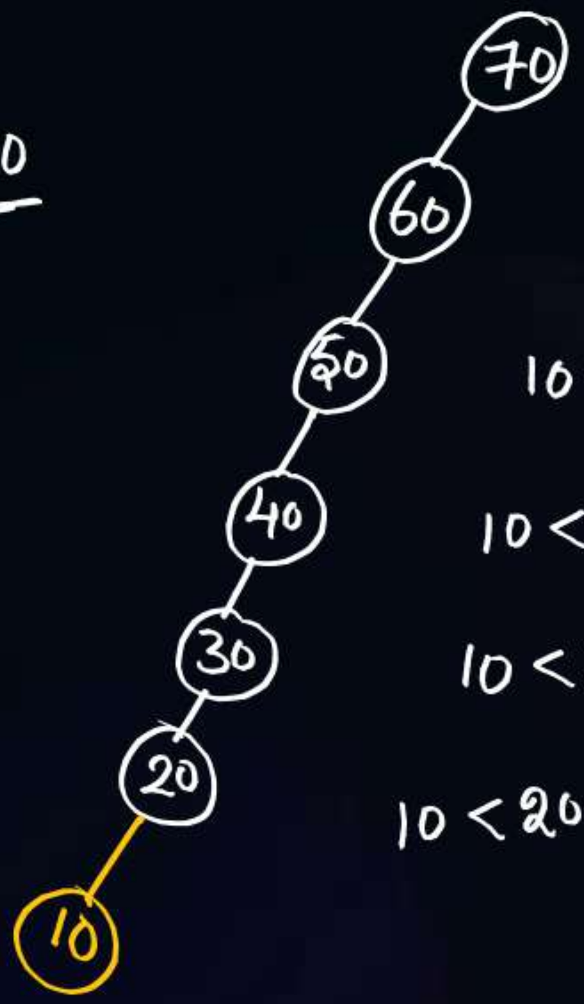
Level order == Pre order

ANS : D

Topic : Binary Search Tree

Insertion Time Complexity

Insert 10



10 < 70

10 < 60

10 < 50

10 < 40

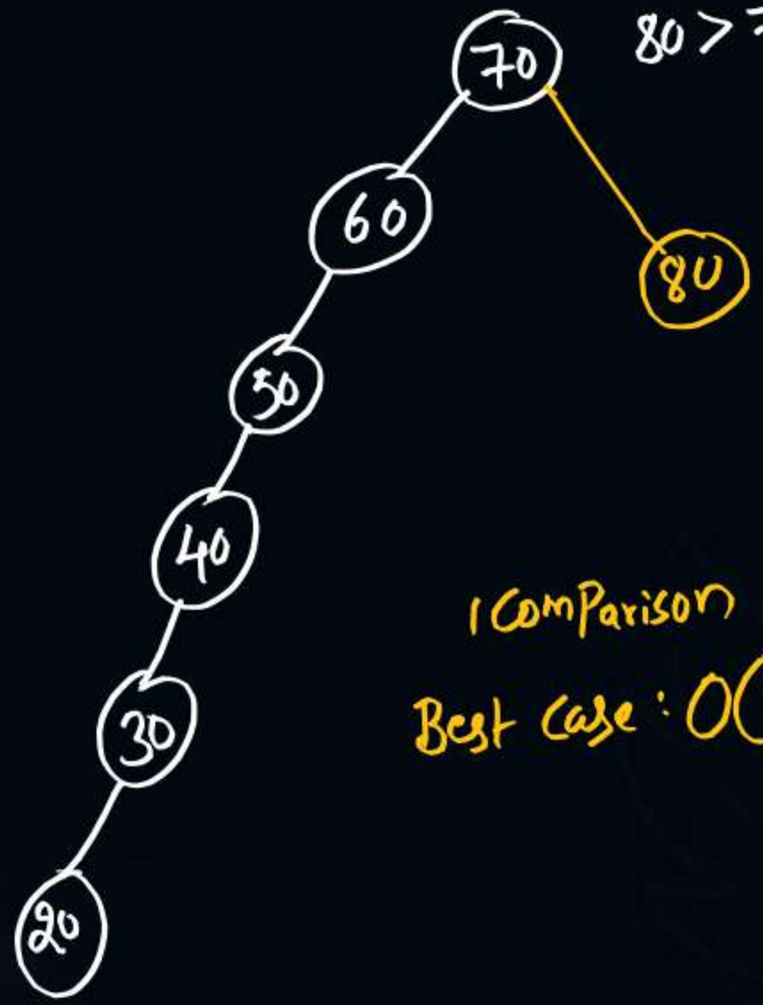
10 < 30

10 < 20

Worst-Case : $O(n)$

(DBT/SBT)

Insert 80

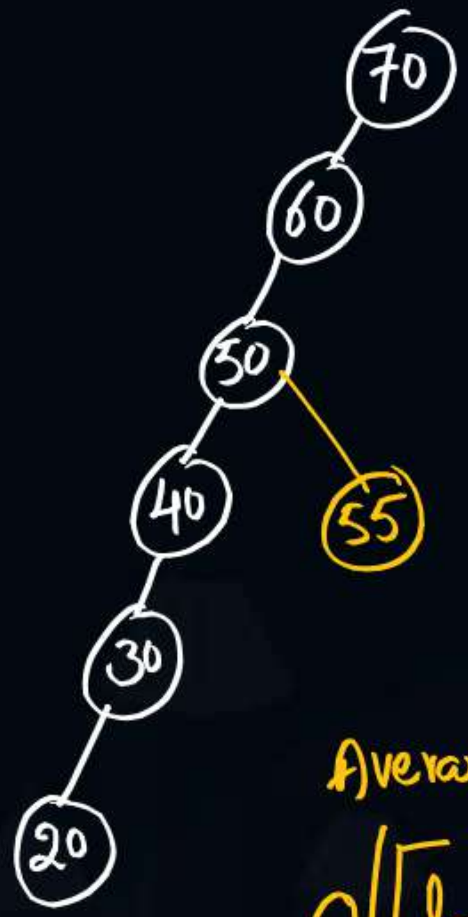


80 > 70

1 Comparison

Best case : $O(1)$

Insert 55



55 < 70

55 < 60

55 > 50

Average Case :

$O\left(\left\lceil \log_2^n \right\rceil\right)$

Topic : Binary Heap - 1

Deletion from BST

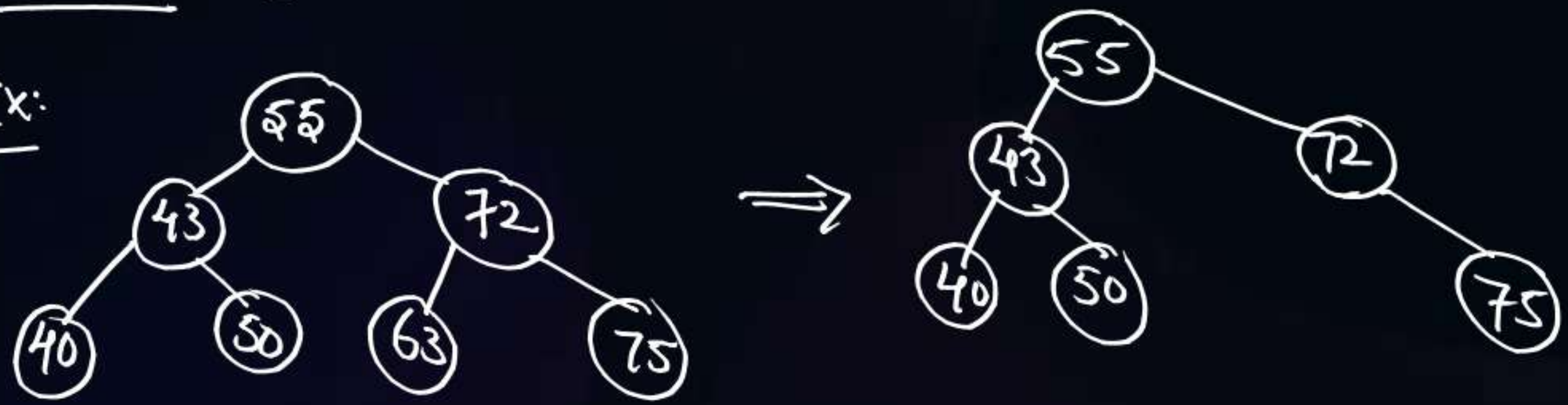
Case-1 : Delete a Node with No children(Leaf Node)

Case-2 : Delete a Node with 1 Child

Case-3 : Delete a Node that has 2 children

Case-1 : Delete a leaf Node : Just Delete it.

Ex:



Let, Delete 63

Topic : Binary Heap - 1

Inorder Traversal of Initial BST : 5, 20, 25, 30, 32, 35, 41, 45, 56, 59, 61, 63, 64, 65

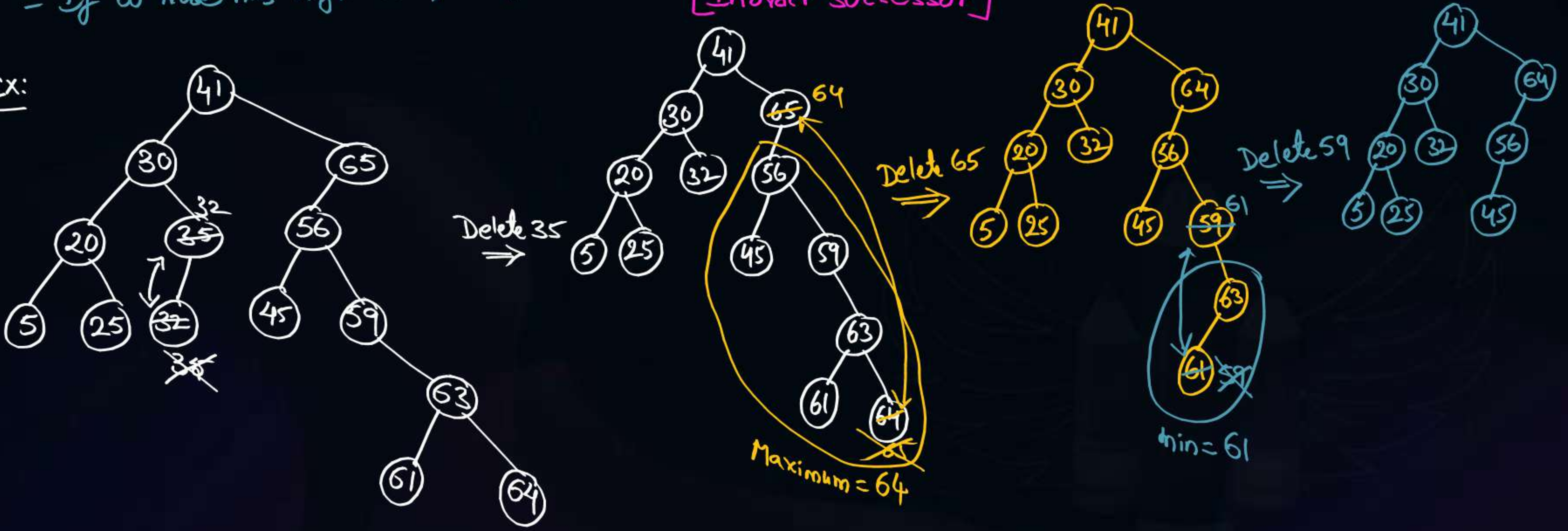
Case-2 Delete a Node that has 1 child

[In order Predecessor]

- If a Node has left child ⇒ Swap Node with Maximum of it's left subtree, Then Delete a Node
- If a Node has Right child ⇒ Swap Node with minimum of it's Right subtree, Then Delete a Node.

[Inorder Successor]

Ex:

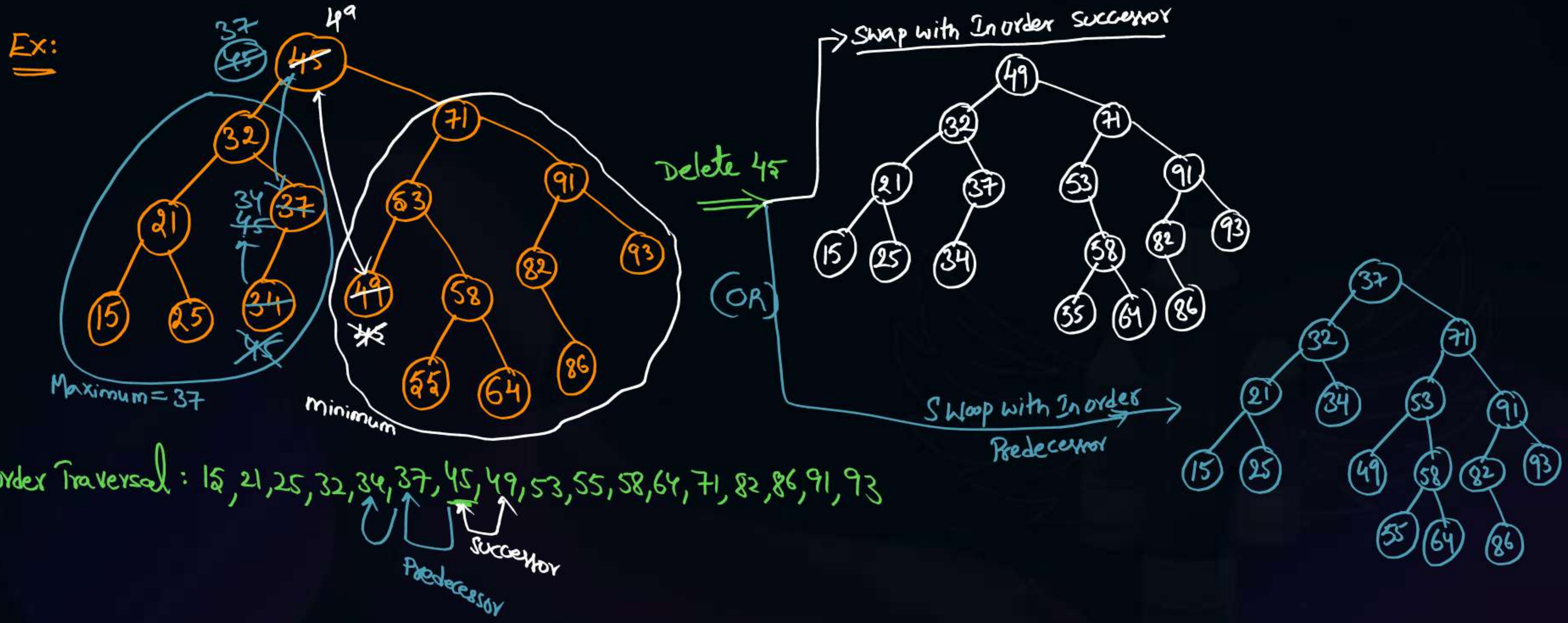


Topic : Binary Heap - 1

Case-3 Delete a Node that has 2 children

Default followed way:

- Swap with Inorder Successor, until Node reaches leaf level
- Swap with Inorder Predecessor, Until Node reaches Leaf Level



Inorder Traversal : 15, 21, 25, 32, 34, 37, 45, 49, 53, 55, 58, 64, 71, 82, 86, 91, 93

Topic : Binary Heap - 1

Binary Heap

- A Binary Tree, Satisfy 2 conditions / Properties :

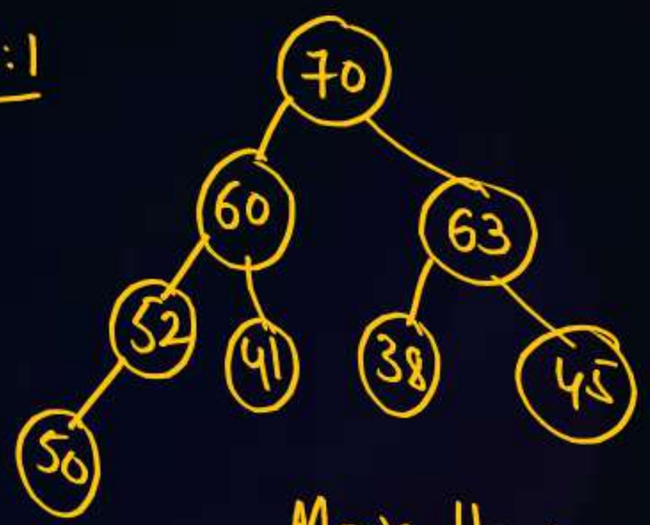
① Complete Binary Tree : Structuring Property / Shape Property

② Every Parent > All children (OR) Every Parent < All children : Ordering Property / Heap Property

Max-Heap

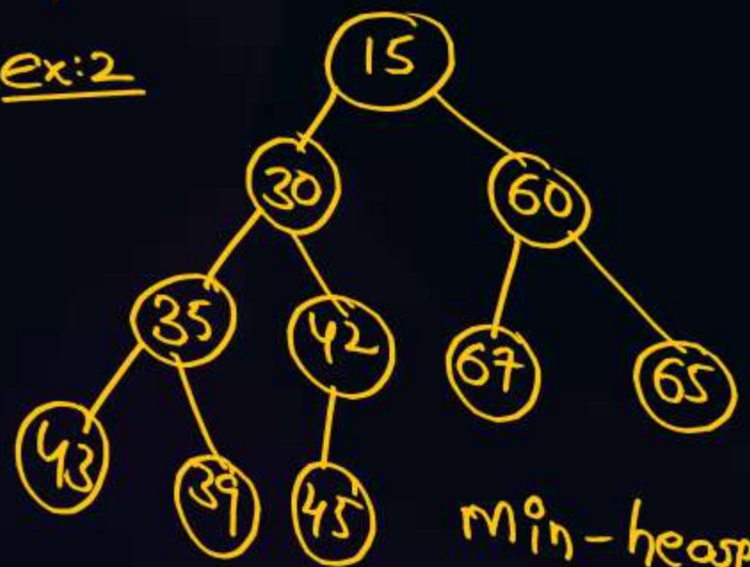
min-heap

Ex:1



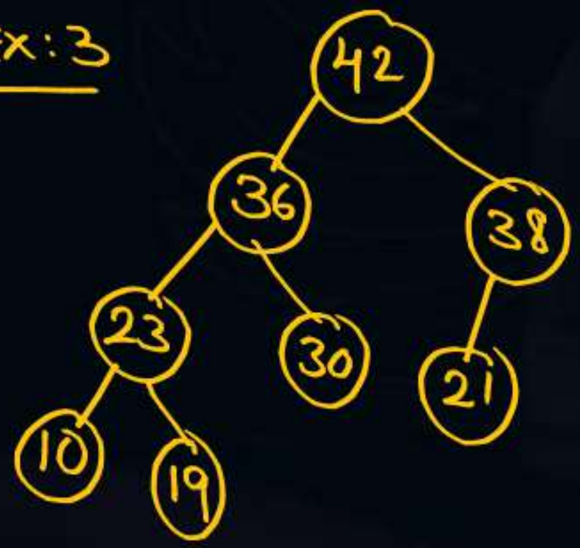
Max-Heap

ex:2



min-heap

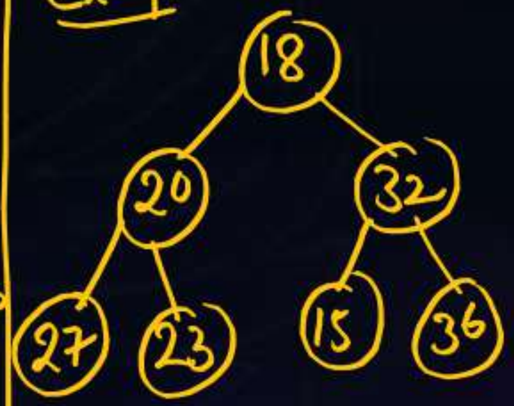
Ex:3



It is Not CBT

Not a Heap

Ex:4



Not a Heap

18>15

Topic : Binary Heap - 1

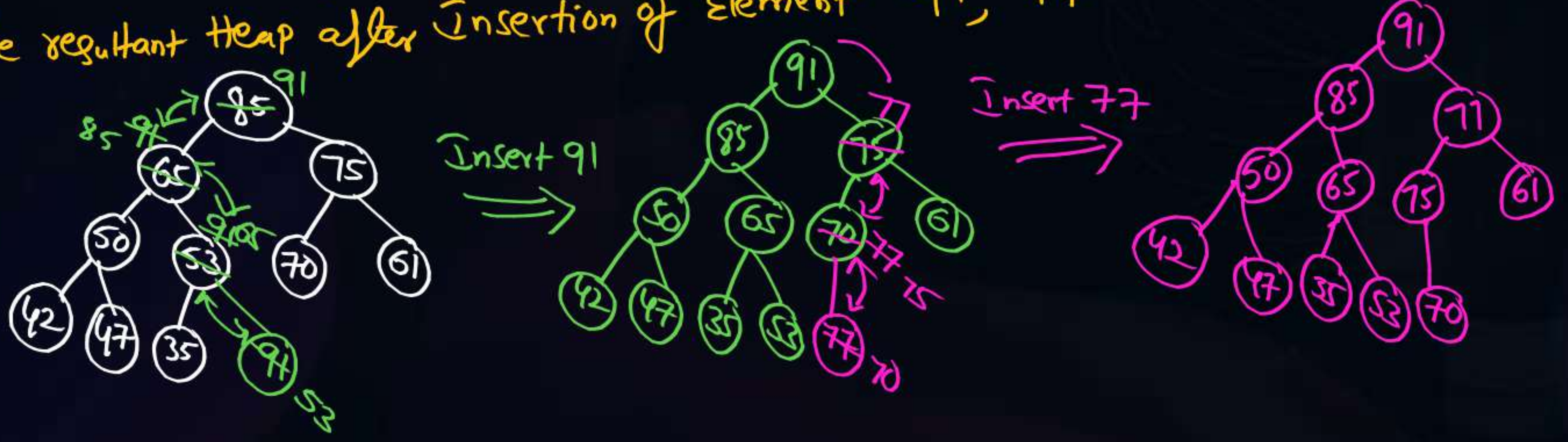
Insertion into Binary Heap

① Insert New Node as last leaf Node

② Heapify : Compare New Node with it's Parent —
- Max-Heap : Parent < New Node, Swap
- min-Heap : Parent > New Node, Swap.

Repeat step ② until all Tree Heapify.

Ex: Consider a Binary Heap with Elements, 85, 65, 75, 50, 53, 70, 61, 42, 47, 35. The resultant Heap after Insertion of Element 91, 77 in that order is ______



Resultant Heap is

91, 85, 77, 50, 65, 75, 61, 42, 47, 35, 53, 70

2 mins Summary

H/w Question

#Q. Consider a Binary Heap, with Elements 15, 20, 23, 31, 36, 42, 39, 45, 51, 60, 70. The Resultant Heap after Insertion of Elements 10, 5, 7 in that Order is ______

THANK - YOU