GATE

CRASH COURSE

Data Science & AI

Subject

Data Structure & Algorithms
BST and Binary Heap
Lecture No. - 05

By – Satya Sir

Last Class

Quick Recap

1. Trees Terminology
2. Types Of Binary Trees
3. Tree Traversals
4. Formulae of Binary Trees

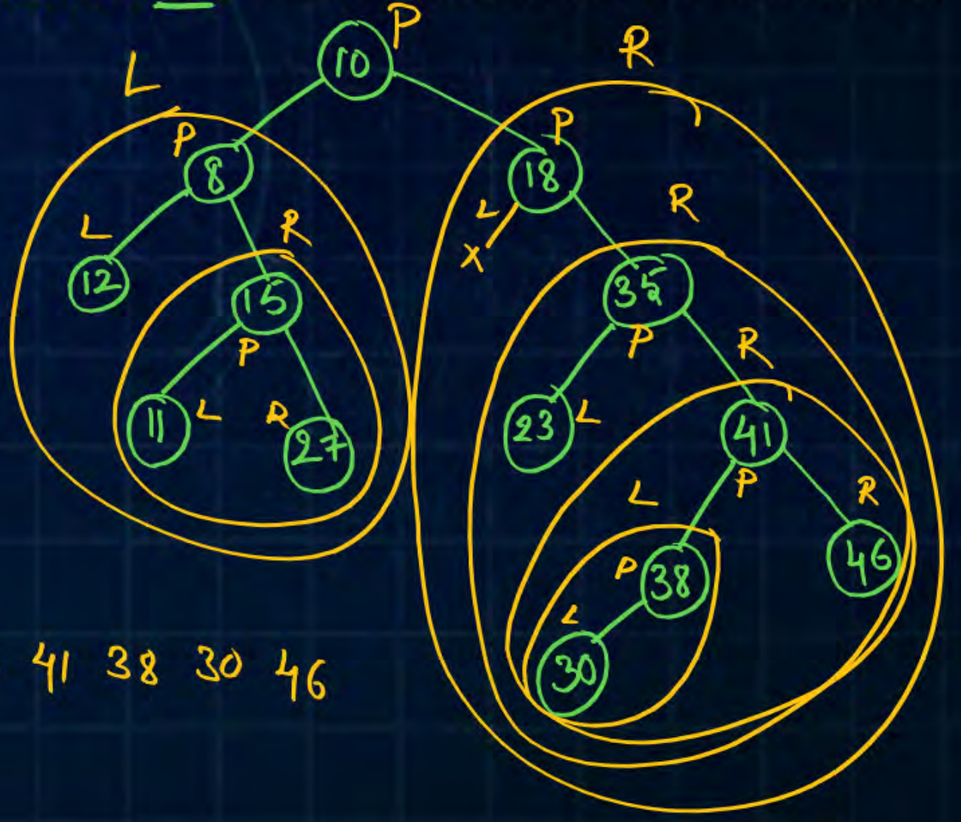
Topics to be covered

1. Homework Problem Solution
2. BST – Insertion, Construction
3. BST - Deletion
4. Binary Heap – Insertion, Deletion
5. Examples

Homework Problem

#Q. Consider the In-order traversal of a binary tree is 12 8 11 15 27 10 18 23 35 30 38 41 46 and Post order traversal is 12 11 27 15 8 23 30 38 46 41 35 18 10. The Pre Order Traversal is _____

a. 10 8 12 15 11 27 18 35 41 23 38 30 46
b. 10 8 12 15 11 27 18 35 23 41 30 38 46
c. 10 8 12 15 11 18 27 35 23 41 38 30 46
d. 10 8 12 15 11 27 18 35 23 41 38 30 46



Pre order Traversal (PLR) : 10 8 12 15 11 27 18 35 23 41 38 30 46

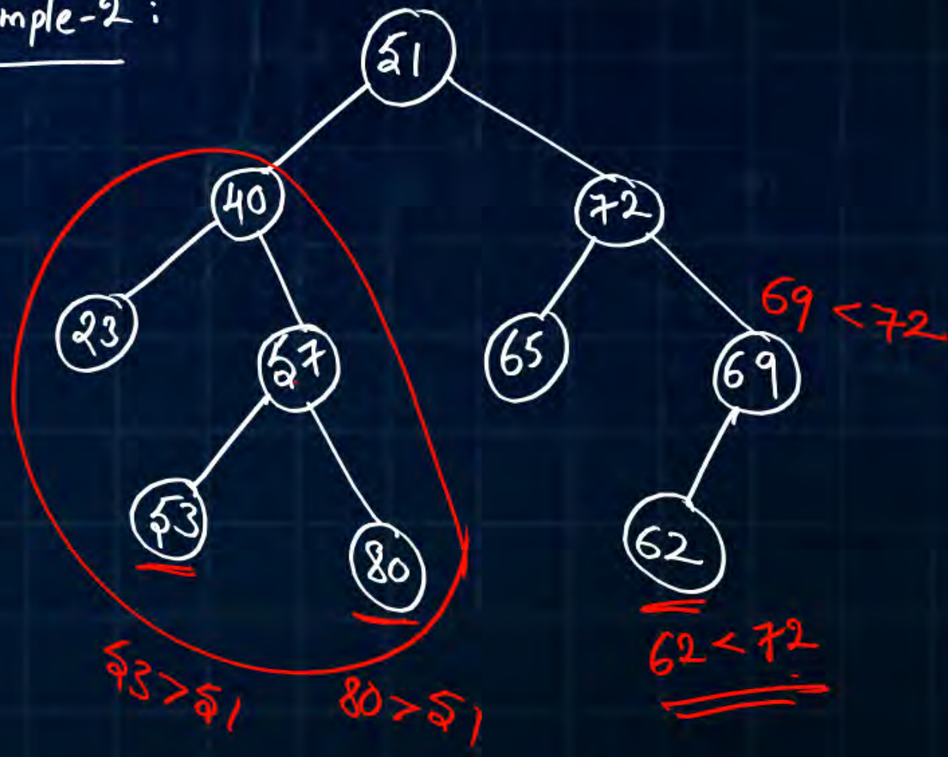
Binary Search Tree

- A Binary Tree, in which **Left Subtree < Parent < Right Subtree** At Each level, is known as BST

Example : 1

Binary Search Tree

Example-2 :

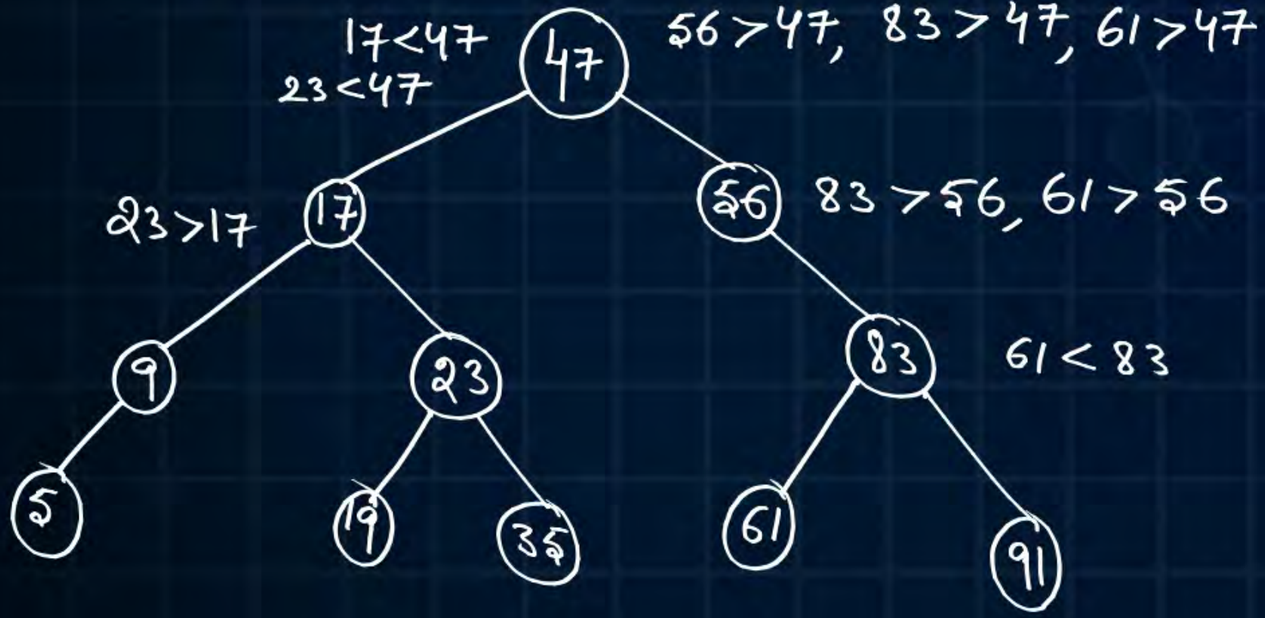


Not a Binary Search Tree

Binary Search Tree

#Q. Construct a BST by Inserting elements, in the order : 47, 56, 83, 17, 61, 23, 19, 9, 5, 91, 35.

The Resultant BST is _____



Resultant BST

= 47, 17, 56, 9, 23, 83, 5, 19, 35, 61, 91

Binary Search Tree

Insertion into BST :

Consider a BST:

Insert 5 : Compare 5,10 5<10

1 Comparision

Time complexity = O(1)

Best-case

Insert 25 : Compare 25,10 25>10

Compare 25,20 25>20

Compare 25,30 25<30

3 Comparisions $= \lceil \log_2^2 \rceil$

Time complexity $\Rightarrow O(\log_2 n)$

Average-case

Insert 55 : Compare: (10,55)

(20,55)

(30,55)

(40,55)

(50,55)

5 Comparisions.

Time complexity = O(n)

Worst-case

Binary Search Tree

Deletion of Element from BST

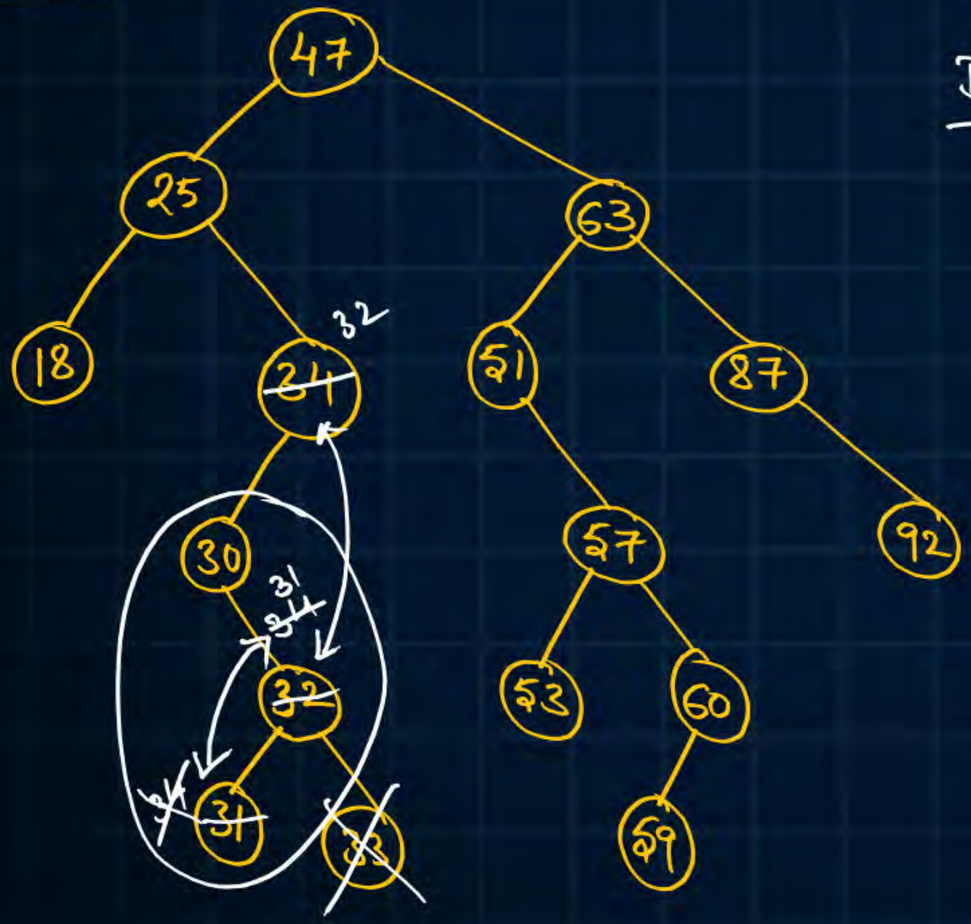
1) Delete a Node with zero children ⇒ Simply Delete.

2) Delete a Node with 1 child
   - → Left child : Swap Node with Inorder Predecessor (Maximum of Left Subtree), Delete
   - → Right child : Swap Node with Inorder Successor (minimum of Right Subtree), Delete

3) Delete a Node with 2 children
   - 1) Swap Node with Inorder Successor
   - 2) Repeat until Node becomes Leaf Node
   - 3) Delete it.

Binary Search Tree

Example :



Delete 33 : Simply Delete it

Delete 34 : Swap [34, Maximum of left Subtree = 32], Swap 34 and 31

Delete 47 (Node with 2 children)

Swap 47, Inorder Successor = 51

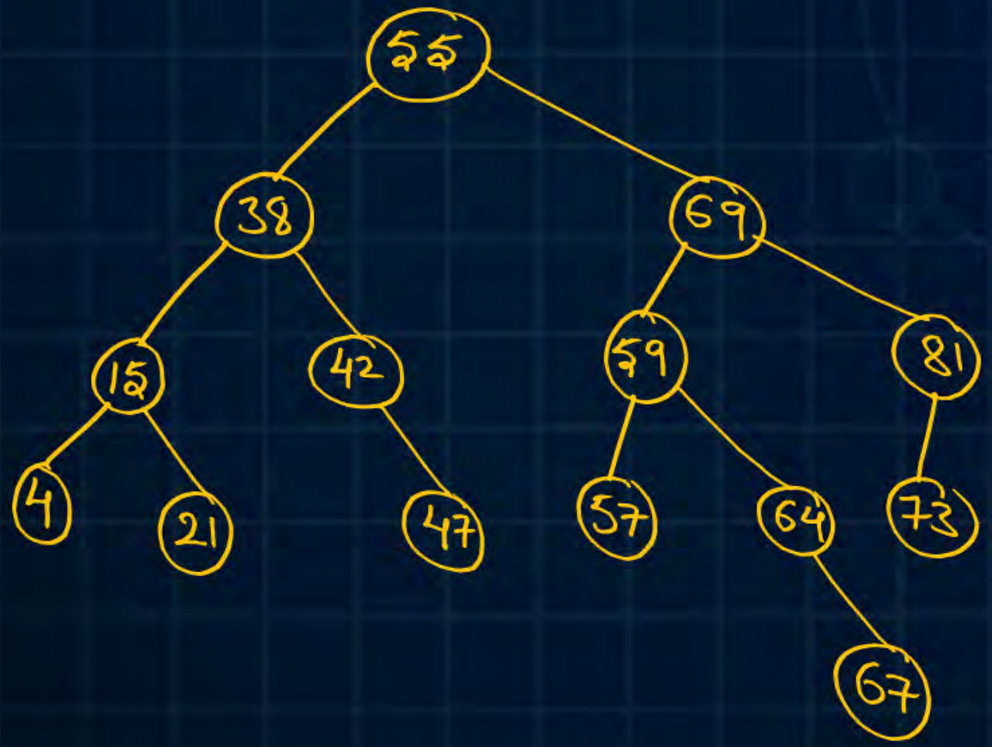
Swap 47, In order Successor = 53

Delete 47

Binary Search Tree

- In order Traversal of a BST will always result elements in the ascending order.

Example :



In order Traversal :

4, 15, 21, 38, 42, 47, 55, 57, 59, 64, 67, 69, 73, 81

Binary Heap

— A Binary Tree, that satisfy 2 conditions below (or) Properties is said to be a Binary Heap.

1) Structuring (or) Shape Property : It must be a Complete Binary Tree.

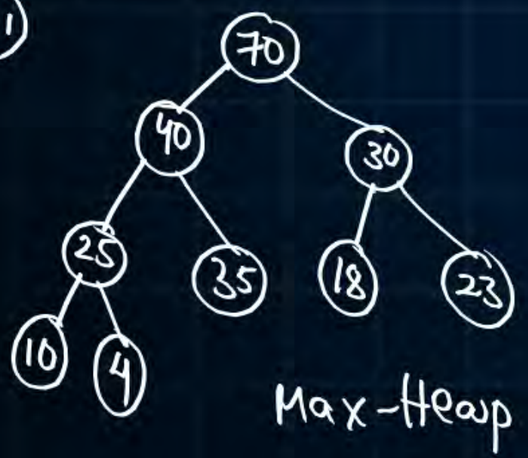
2) Ordering (or) Heap Property : Each Parent Value > All children Values ⇒ Max-Heap

(OR)

Each Parent Value < All children Values ⇒ min-heap.

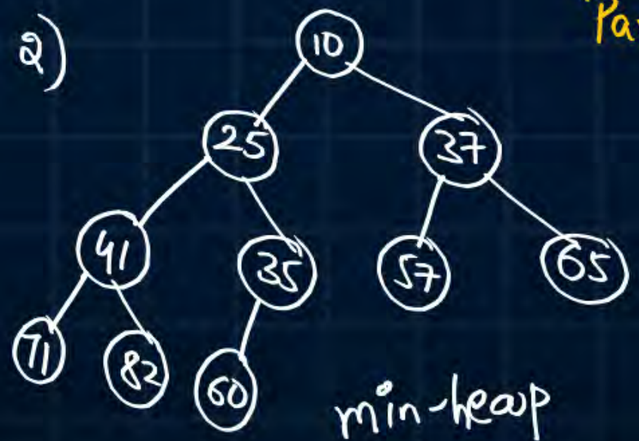
Examples :

1)



Max-Heap

2)



min-heap

3)

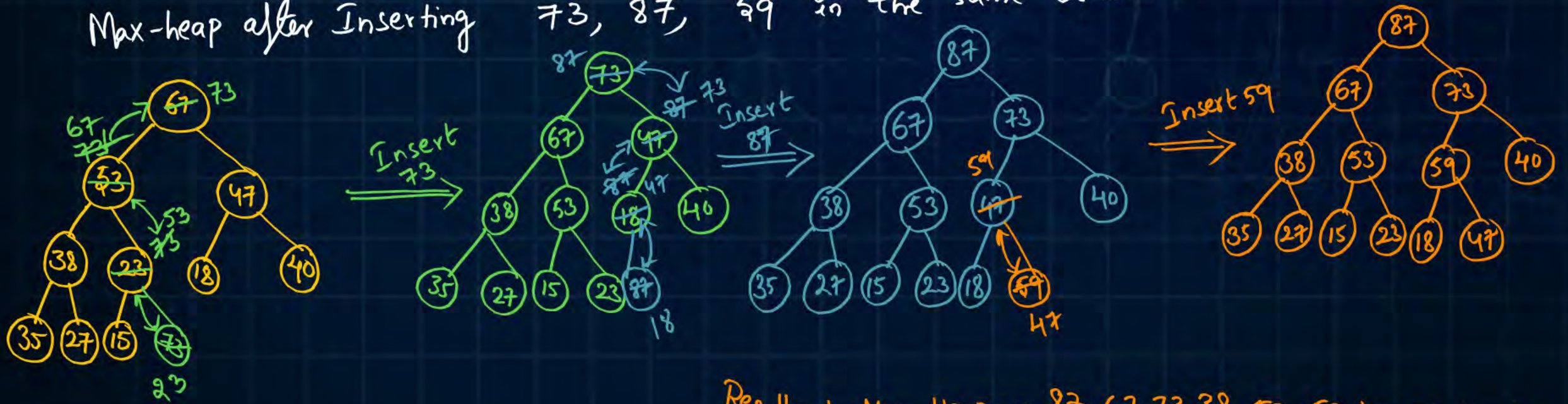
Not a CBT

⇒ Not a Binary Heap.

Binary Heap

Insertion into Heap

#Q. Consider a Max-Heap with elements 67, 53, 47, 38, 23, 18, 40, 35, 27, 15. The resultant Max-heap after Inserting 73, 87, 59 in the same order is _____



Resultant Max-Heap == 87, 67, 73, 38, 53, 59, 40, 35, 27, 15, 23, 18, 47

Binary Heap

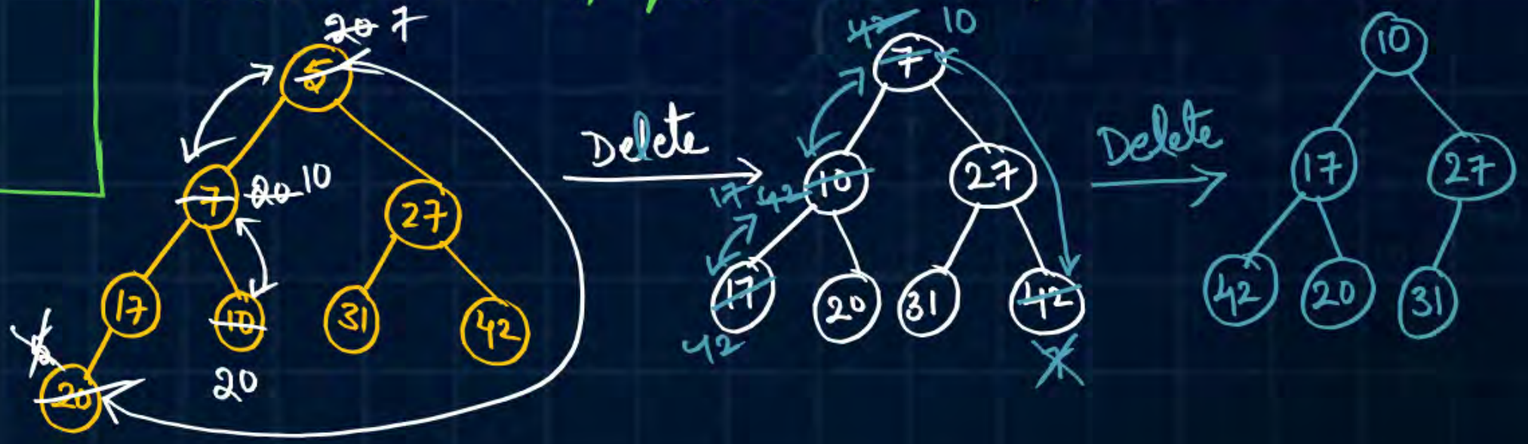
Deletion from Binary Heap :

- Deletion in a Heap by default Performed at root Node only, To Preserve CBT Property.

1) Swap root Node with last leaf
2) Delete Node
3) Heapify (Compare and Swap).

Ex: Consider a min-heap, with Elements : 5, 7, 27, 17, 10, 31, 42, 20.

The Resultant min-heap, after 2 Deletion operations is _____



The Resultant min-heap : 10, 17, 27, 42, 20, 31

Binary Heap

Home-work

#Q. Consider a Max-heap with Elements: 85, 72, 63, 57, 42, 60, 38, 43, 19. The Resultant Max-Heap after following operations, in the same order is _____

1) Insert 80
2) Delete
3) Insert 93
4) Delete

Post Your Homework Answers / Queries / Doubts @

t.me/ satyasirpw

Summary

- Binary Search Tree
  - Construction
  - Insertion
  - Deletion
- Binary Heap
  - Insertion
  - Deletion


Thank
THANK
Keep Hustling!