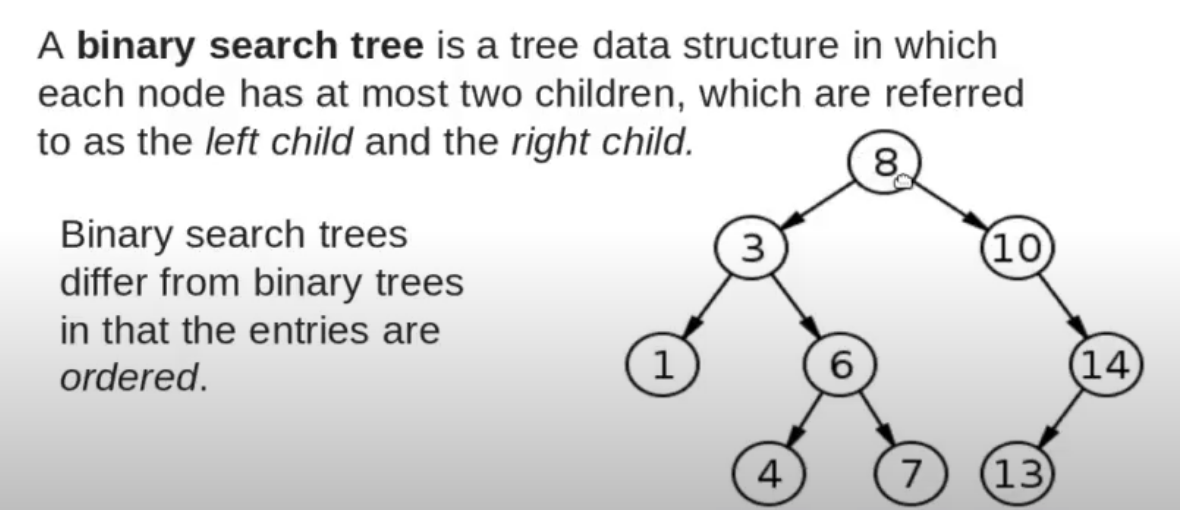
# **Difference**

Graphical user interface, text, application

Description automatically generated



# **Data Insertion**

## **Example-1 (Given -> Elements are not sorted)**

Ask question to every node,   
if elements found to be lesser 🡪 go to left

if elements found to be greater 🡪 go to right

Graphical user interface

Description automatically generated with medium confidence

If Element-8 is less than 3, insert the element to its left (or) insert the elements to its right.

Inserting element-3 to 8’s left

Graphical user interface, application

Description automatically generated with medium confidence

Inserting element-10 to 8’s right



Inserting element-1.  
a) 1<8, so left  
b) 3<8, so left

Similarly for element-6

Diagram

Description automatically generated with medium confidence

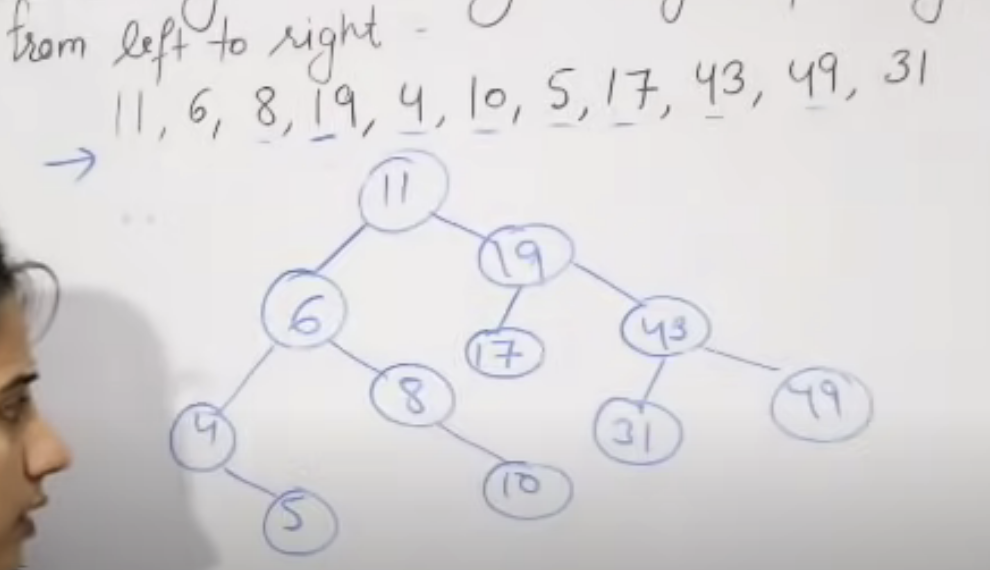
## **Example-2 (Given -> Elements are in reverse-sorted)**

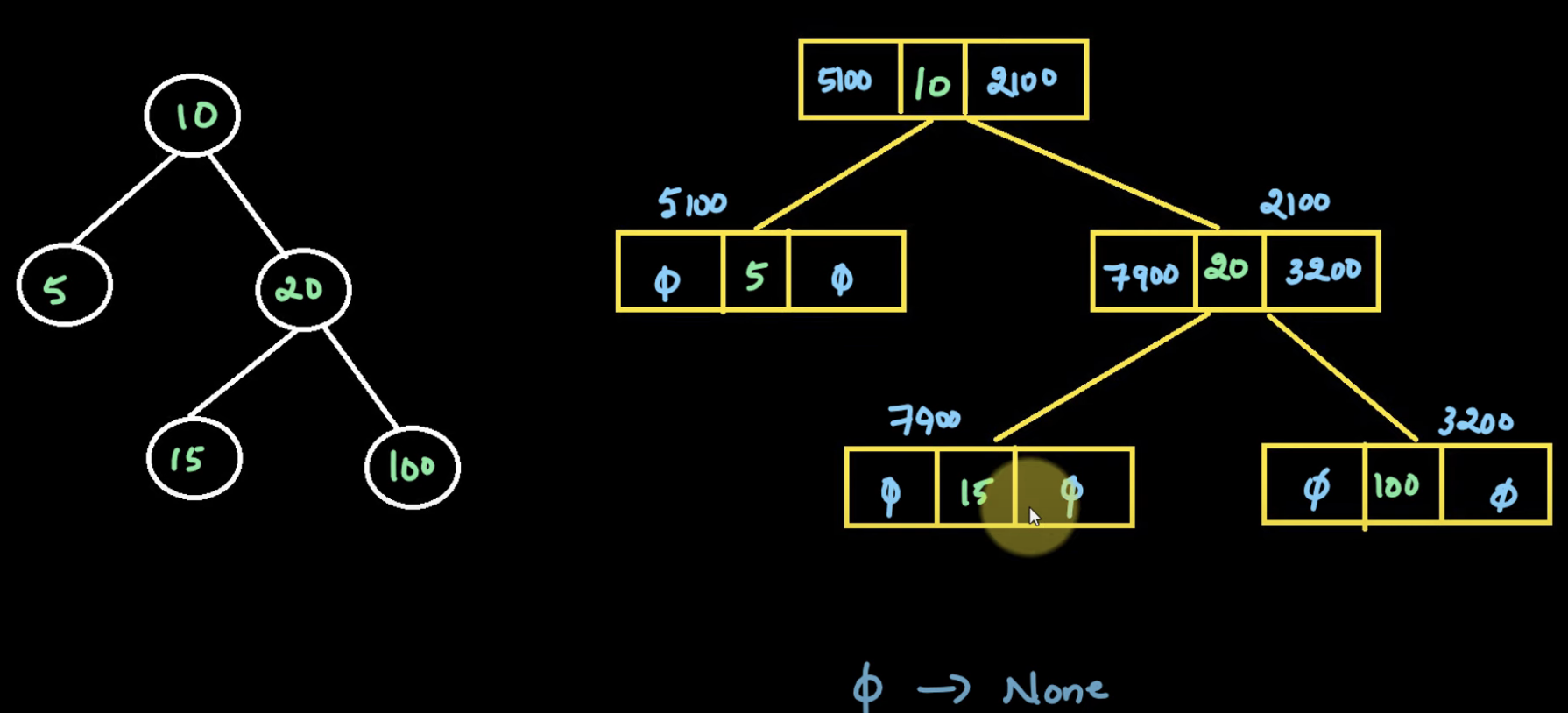
Background pattern

Description automatically generated with low confidence

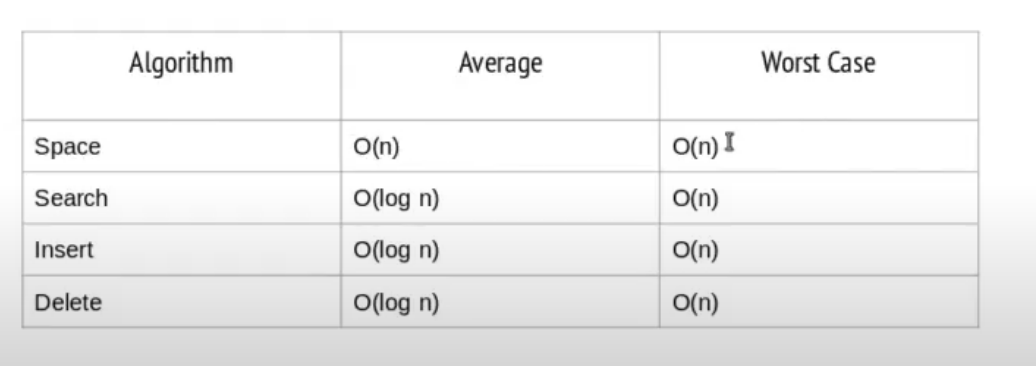
A picture containing diagram

Description automatically generated





# **Data Search**



# **Data Deletion**

## **Deleting a Node with 0 child**

🡪 Simply Delete that node

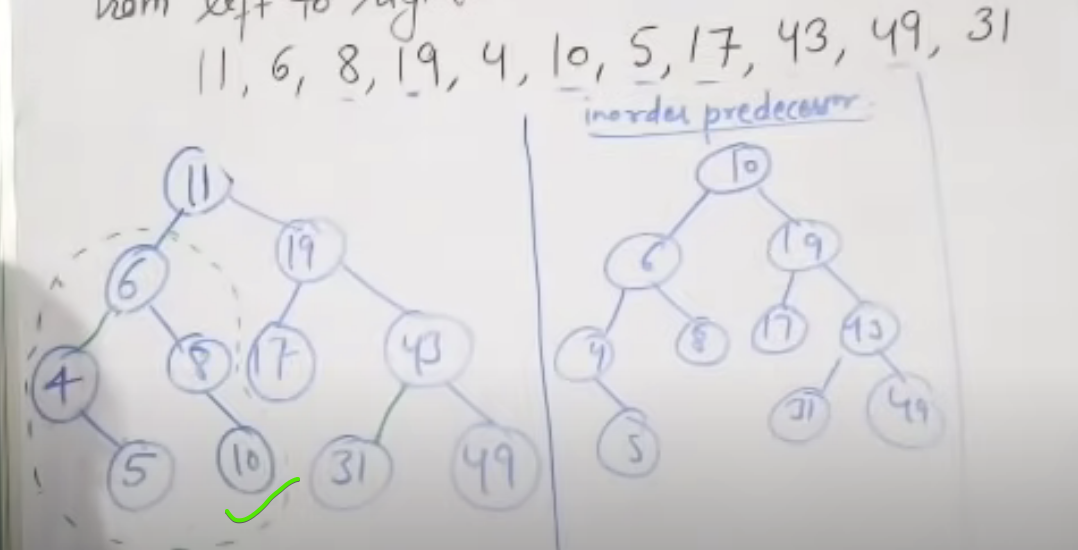
## **Deleting a Node with 1 child**

Delete the Node-4 that has only 1 child (Node-5).  
That particular node will be replaced by its child (whether left or right child)

## **Deleting a Node with 2 children**

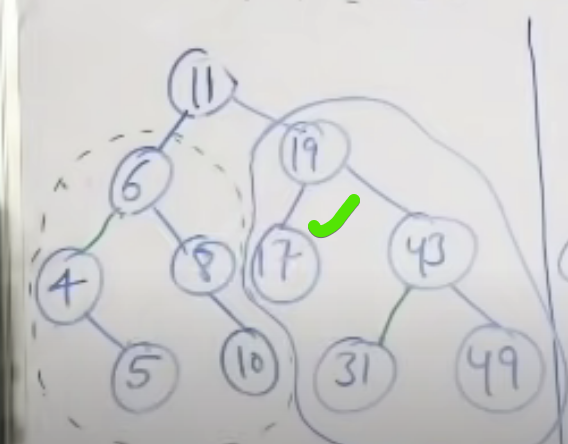
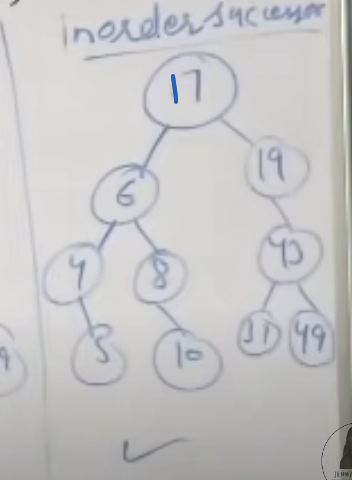
### **Replace that node with its in-order predecessor**

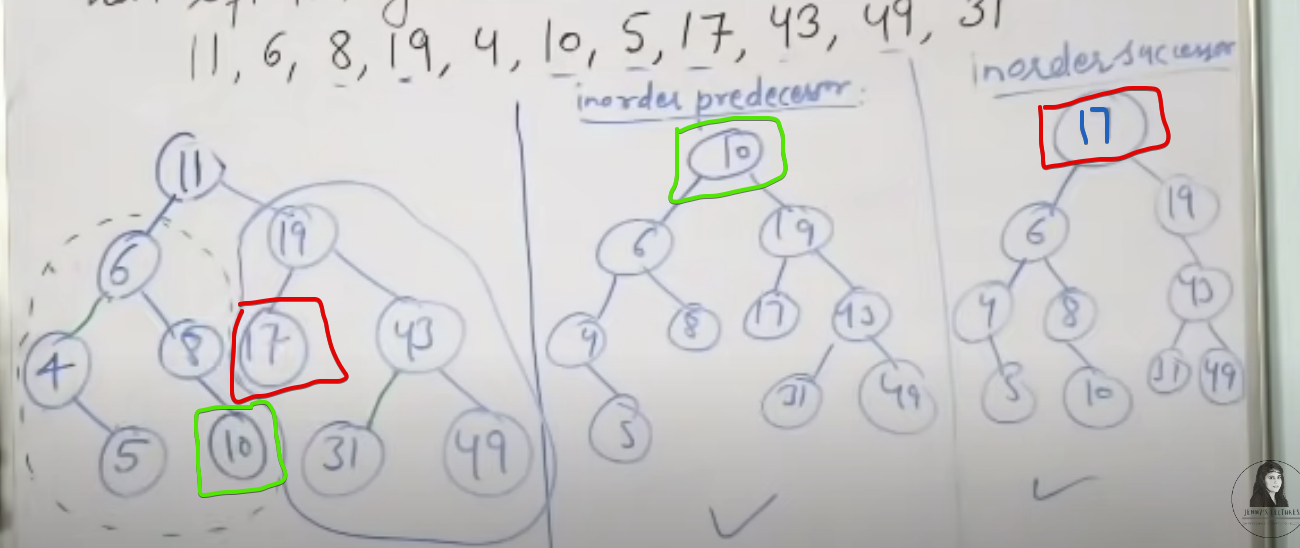
Largest element from its left sub-tree, Here the largest is 10.



### **Replace that node with its in-order successor**

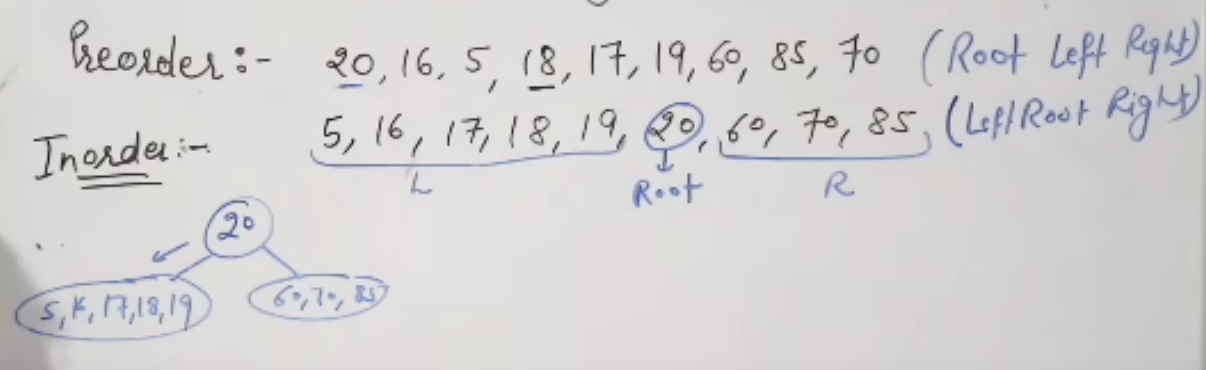
Smallest element from its right sub-tree, Here the largest is 17.

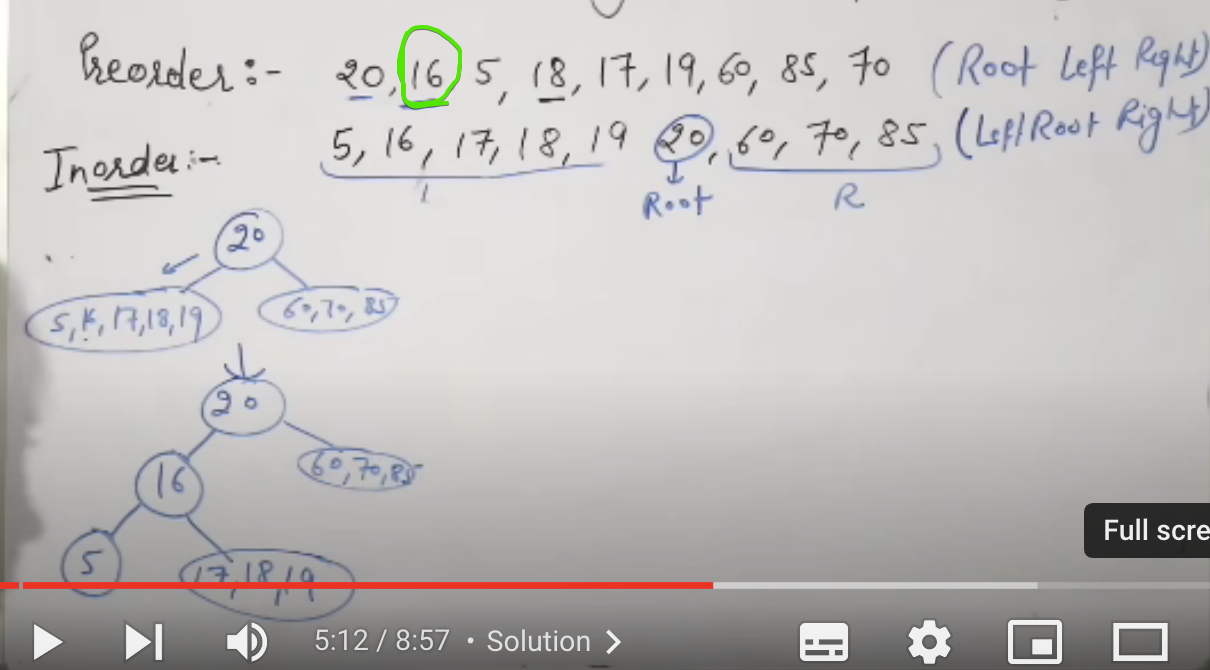
 

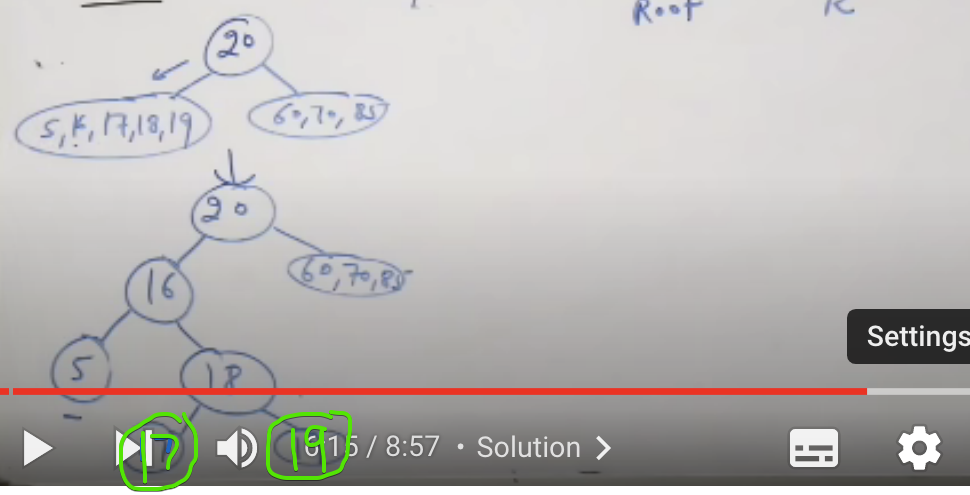


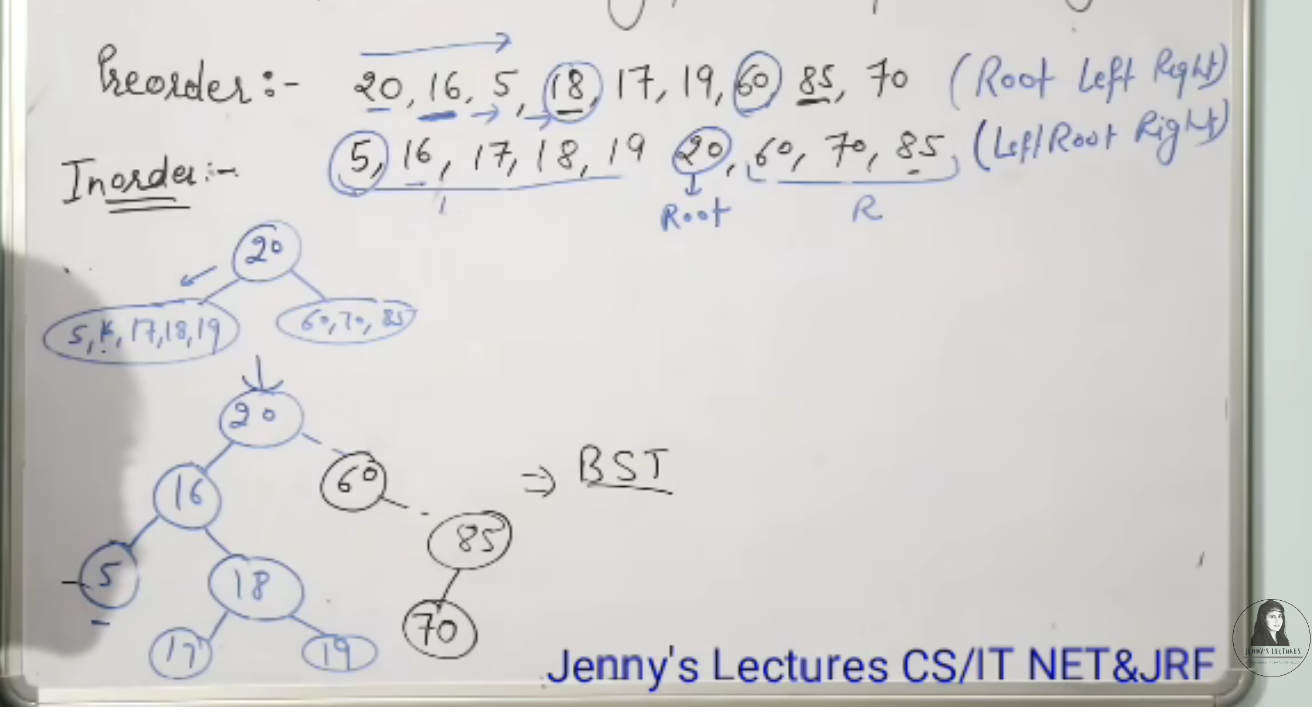
Both the in-order predecessor and in-order successor will be in sorted order.

# **Construction from Pre-Order**

  
Number which is coming 1st in the pre-order will be the root.

Since 16 is coming first, to the left sub-tree Node-16 will be the root.  


Left 🡪 5 is coming first, so it is the root node  
Right 🡪 18 is coming first, so it is the root node  


Final Tree  


# **Construction from Post-Order**

# **Reference**

<https://www.youtube.com/playlist?list=PL5tcWHG-UPH112e7AN7C-fwDVPVrt0wpV>