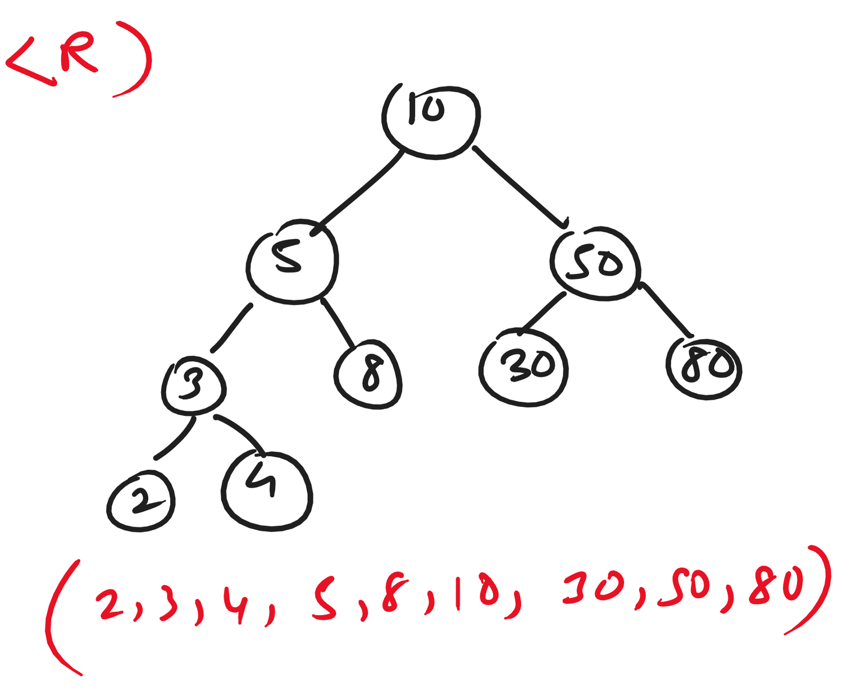
BST Implementation :→ (L<N<R)

Operations :→

1. Insert (root, key)
2. Search (root, key)
3. Delete (root, key)
4. InOrder Traversal
   ↳ Ascending Order



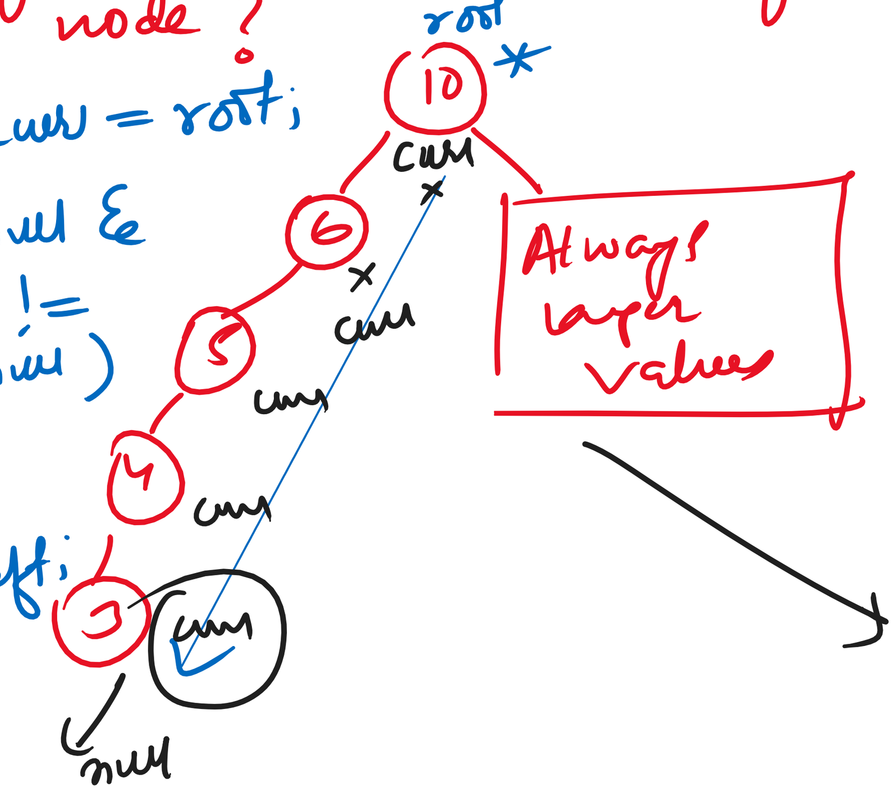
(2, 3, 4, 5, 8, 10, 30, 50, 80)

Note: For (skewed data) the BST height becomes O(n) or O(h) where n or h becomes the height of the tree.
Therefore, we use AVL Trees or Red Black Trees for strictly height balanced BST property.
Balance Factor = height(left) − height(right)
(0 to ± 1)

[1962] → Thesis Report → (Process of Storing Infor")
Adelson Velskey & Engenii Landis
AVL (Rotations)

* How to find the least value of a BST for any given node?

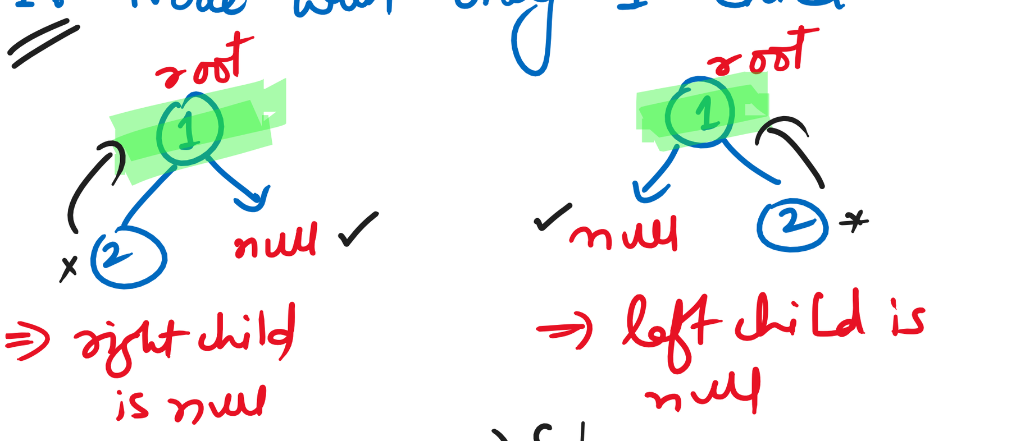
TreeNode curr = root;
while (curr != null & curr.left != null)
{
curr = curr.left;
}
return curr;



x x findMin (root)
x findMax
r.right;
curr

Three important factors for the delete operation in a BST.

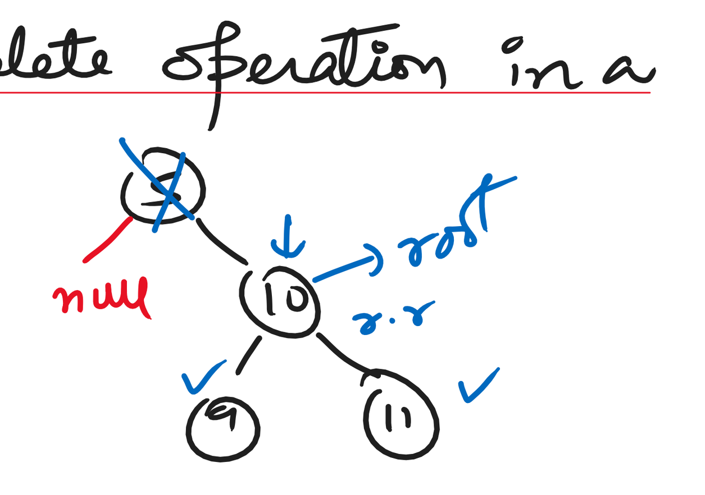
1. Node with only 1 child

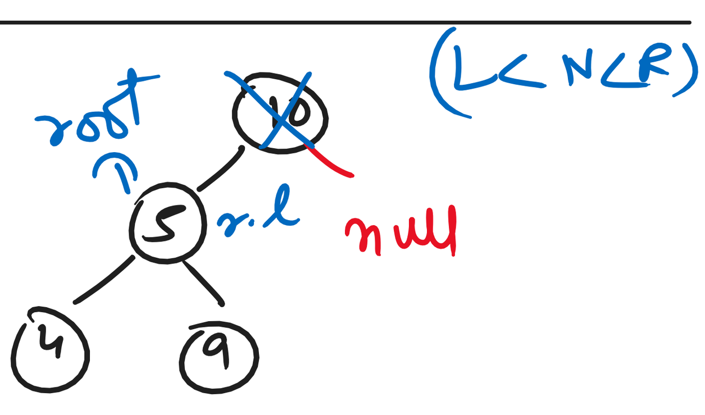


⇒ right child is null
⇒ left child is null

```
if(root.right==null){
  Tree temp = r.l
  root = null;
}  return temp;
```

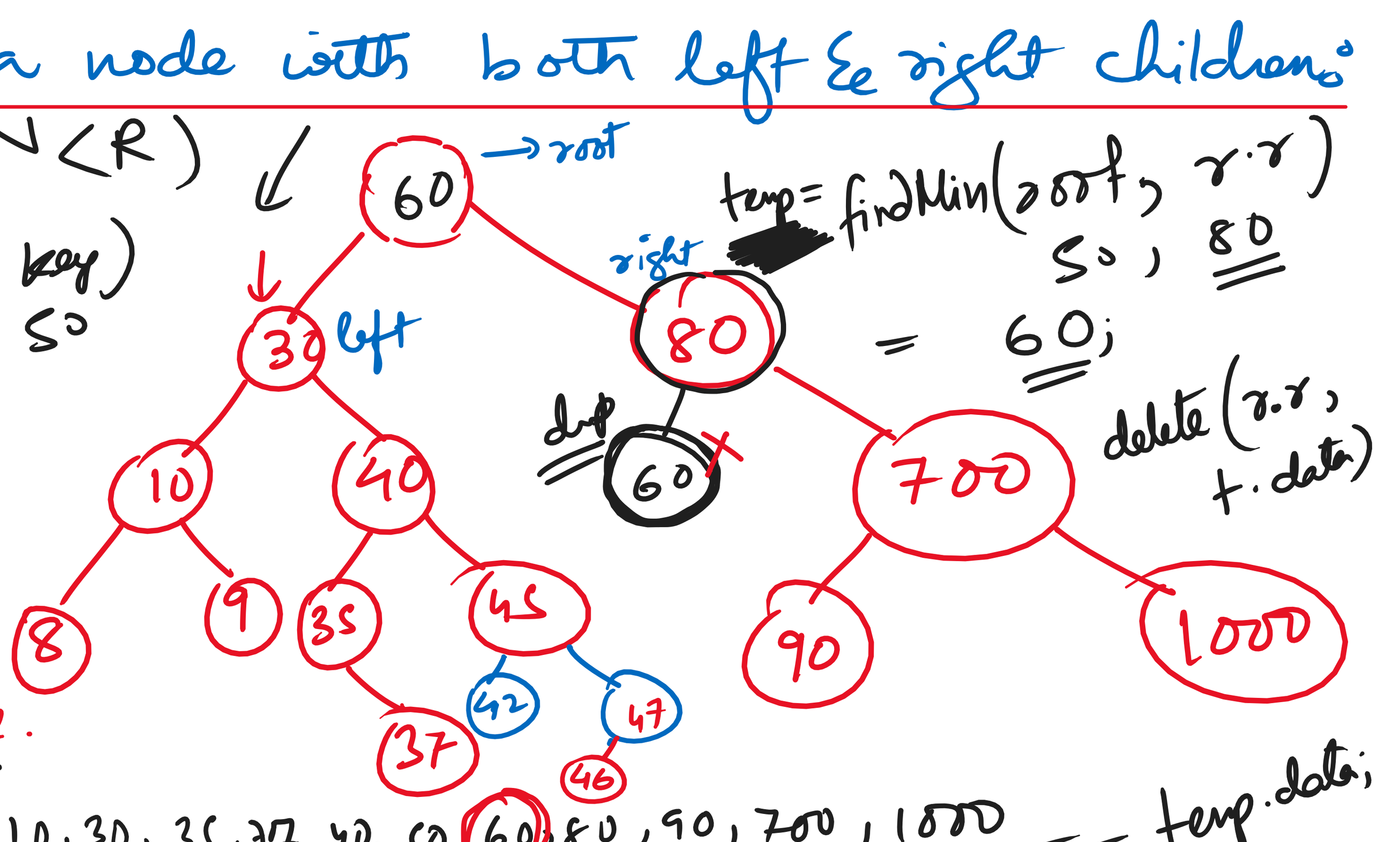
```
Tree temp = r.r
root = null;
return temp;
```





* Delete a node with both left & right children:

BST (L<N<R)
delete (root, key)
50, 50



temp = findMin(root, r.r)
50, 80
= 60;
delete (r.r, t.data)

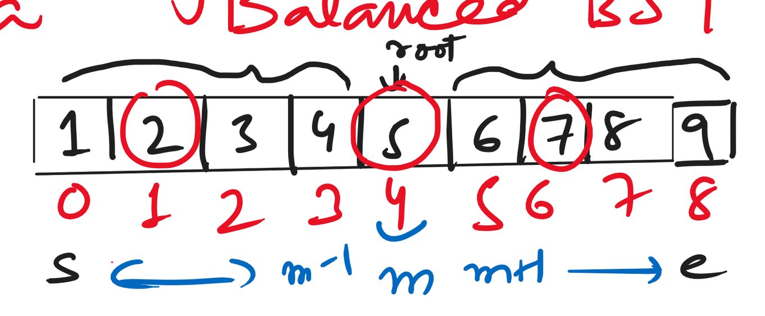
1 problem if you consider left side: more restructure.

In-order: 8, 9, 10, 30, 35, 37, 40, 50, 60, 80, 90, 700, 1000
42, 45, 46, 47

r.data = temp.data;

* [In-order Successor] to replace the root

* Write a java code to convert a sorted array to a Balanced BST:

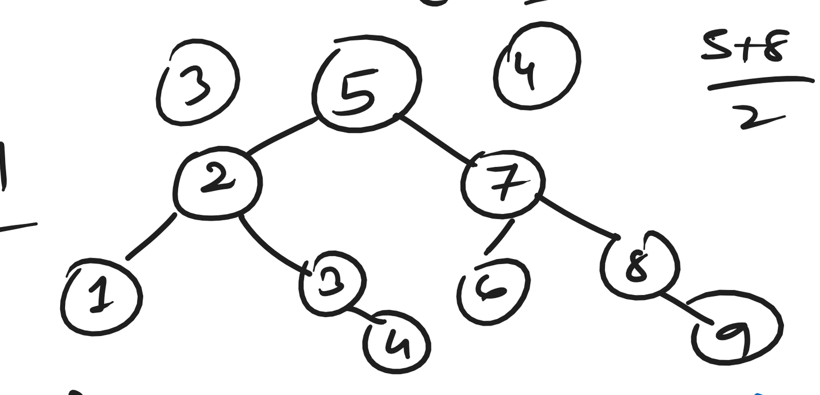


$m = s + \frac{(e-s)}{2}$ / $\frac{s+e}{2} = \frac{8}{2} = 4$

recursion root = arr[mid] recursion

$\frac{3}{2} = 1$

0, ±1



$\frac{5+8}{2} = \frac{13}{2} = 6$

3 − 4 = −1

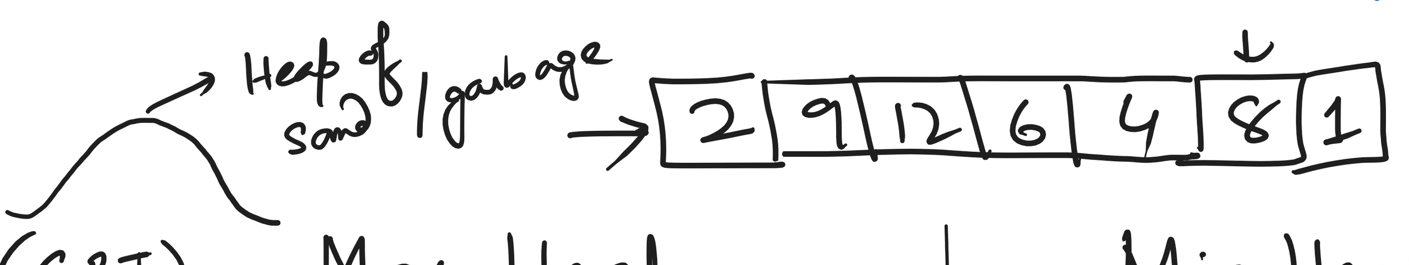
(7:30 PM)
Cognizant → August 4th 2025
Wipro → September 27th 2024
Capgemini → March 30th 2025
Intel → April → 2025
Bosch → May → 2025
HP → 2022
IBM → 2023

Types of trees :→ (GFG)

(Algorithm)
↓
* Abdul Bari
* Kunal Kushwaha

0 / 1 / 10 LPA:
1. Normal Trees
2. Binary Trees
3. Search Trees → BST → Not height balanced; Right / Left > skewed trees
* Complete Binary Tree (Heap)
→ AVL →
→ Red Black → } Strictly Height Balanced (Use rotations for balancing)
Balance factor = lh − rh
range [0 to ± 1]

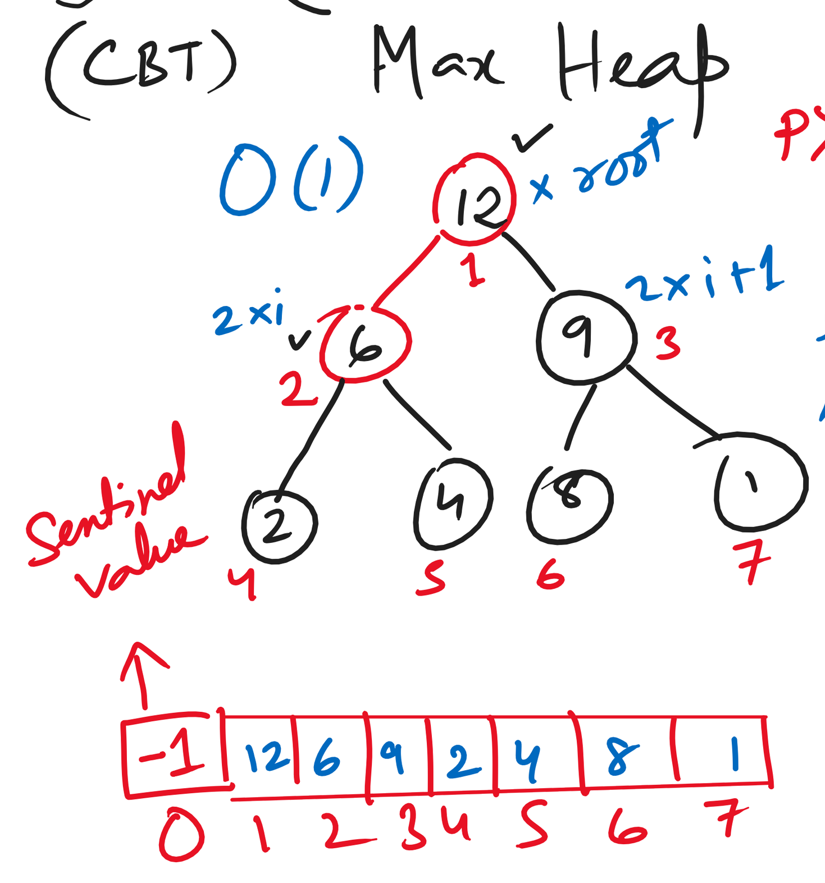
10+ LPA:
4. Fenwick Trees (Binary Index Tree)
5. Segment Trees
6. K-dimensional Trees
7. Orthogonal Range Trees
8. B / B+ Trees (DBMS) (Database)
9. Suffix Trees → Tries → Phone Book or Directory → word Dictionary



O(1) TC
constant time

(CBT) Max Heap | Min Heap

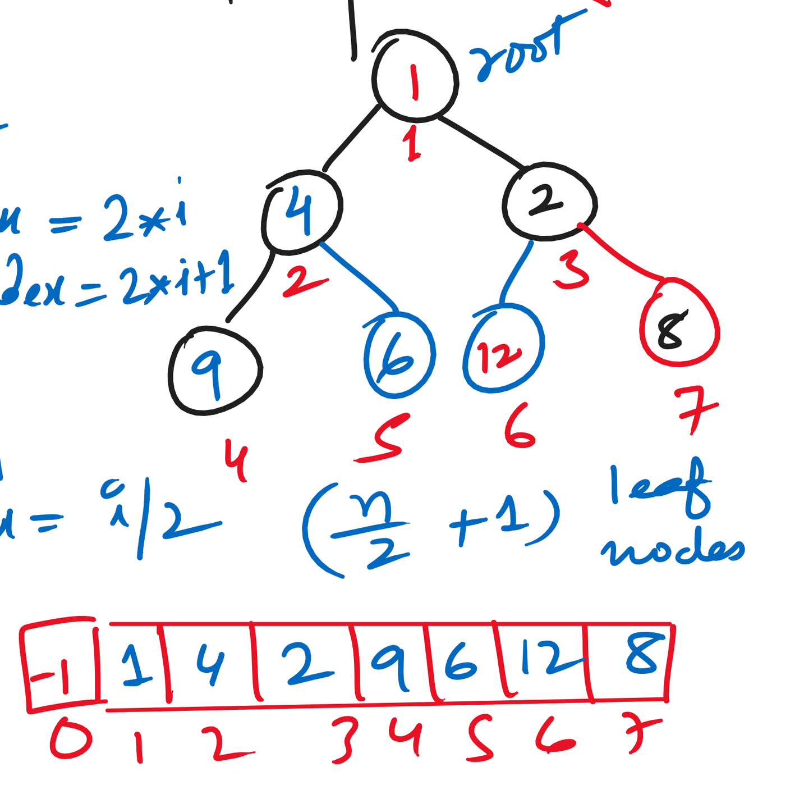
O(1)
P > l & r



i → parent
left child index = 2*i
right child index = 2*i+1

i → child
parent index = i/2

P < l & r



$(\frac{n}{2} + 1)$ leaf nodes