Case 1
- Leaf Node → return node
- Only having left child → return node.left
- Only having right child → return node.right
- Both children present
  - search for max val in left subtree
  - put that value on node val
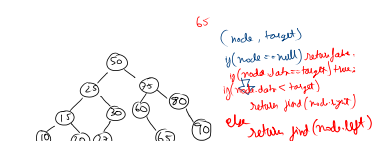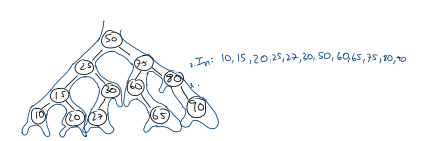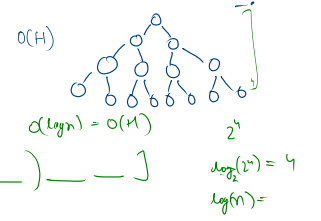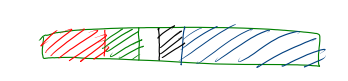 - delete node(node.left, maxval)

BST

1) For every node, the left subtree will have all values strictly smaller than nodeval
   right will have all values strictly larger than nodeval

2) Inorder traversal will be sorted

In: 10, 15, 20, 25, 22, 20, 50, 60, 65, 75, 80, 90

$O(1) = \log(n)$

$O(H)$

(node, target)
if(node == null) return false
if(node.data == target) true
if(node.data < target)
    return find(node.right)
else return find(node.left)

$O(\log n) = O(H)$

$\log_2(2^4) = 4$

$\log(n) =$

4294967296

2^{32}