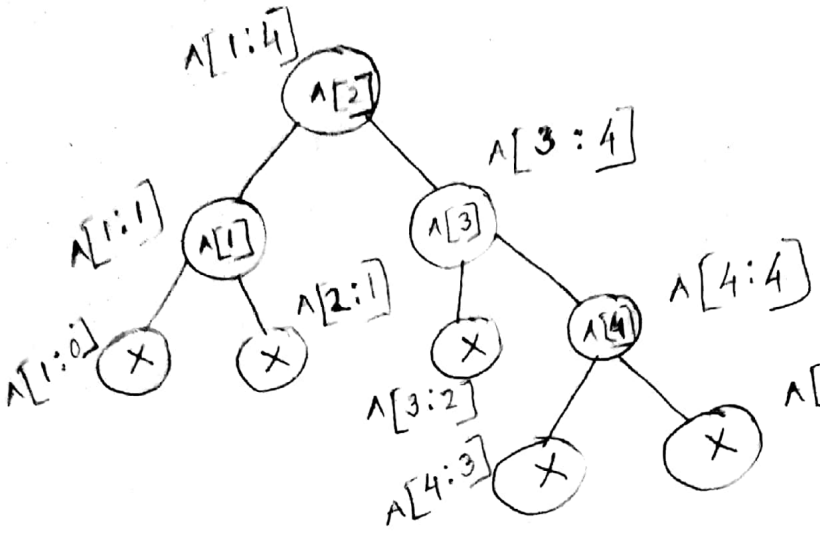
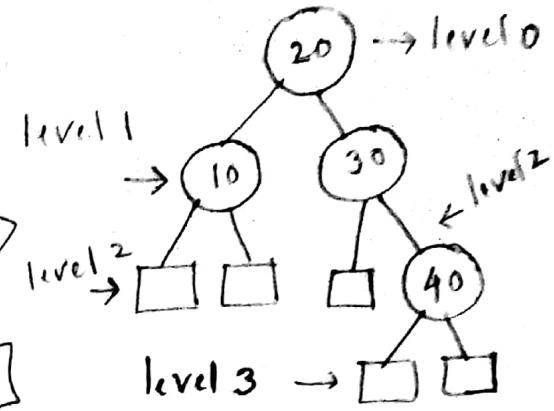
# Binary Search Tree

Draw a binary search Tree for A[1:4]

Ans:

| A | 10 | 20 | 30 | 40 |
|---|---|---|---|---|
| | 1 | 2 | 3 | 4 |




Binary Search Tree

I(T) = Internal Path length

= 0+1+1+2 = 4

E(T) = External Path length

= 2+2+2+3+3 = 12

E(T) - I(T) = 2n

⇒ 12 - 4 = 2*4

I(n) = ○ = internal node

E(n) □ = External node

E(n) = I(n) + 1

⇒ 5 = 4+1

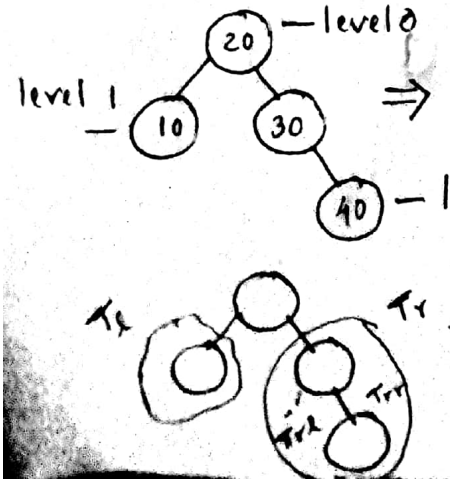
Height of a binary search Tree

Tree = NULL height = -1

○ single node, height = 0

otherwise, height(Tree)

= max(height(Tl), height(Tr)) + 1




* height(Tree) = max(level i) = 2

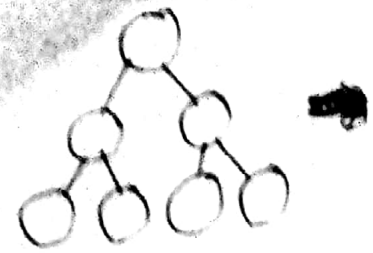
* height(T) = max(height(Tl), height(Tr)) + 1

= max(0, max(height(Tll), height(Trr)) + 1) + 1

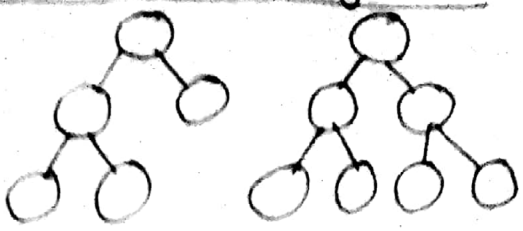
= max(0, max(-1, 0) + 1) + 1

= max(0, 0+1) + 1 = 1+1 = 2

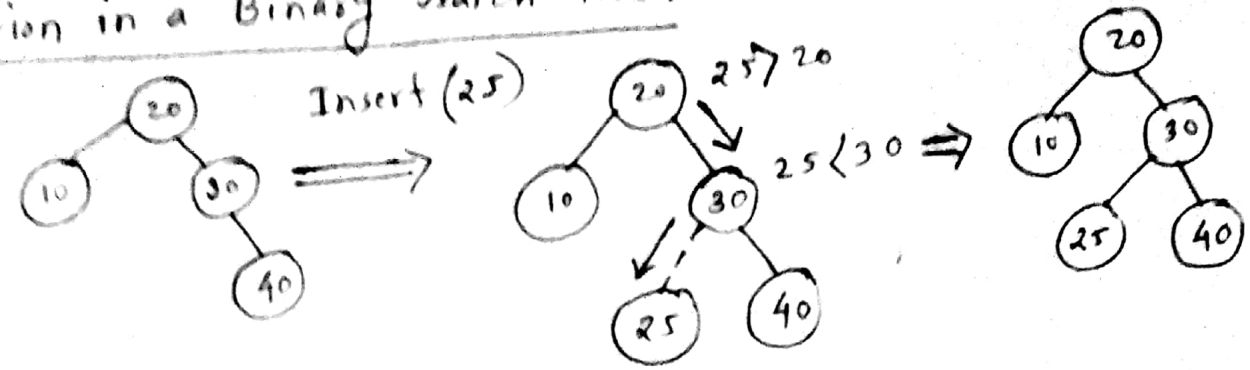
Balanced Binary Tree:

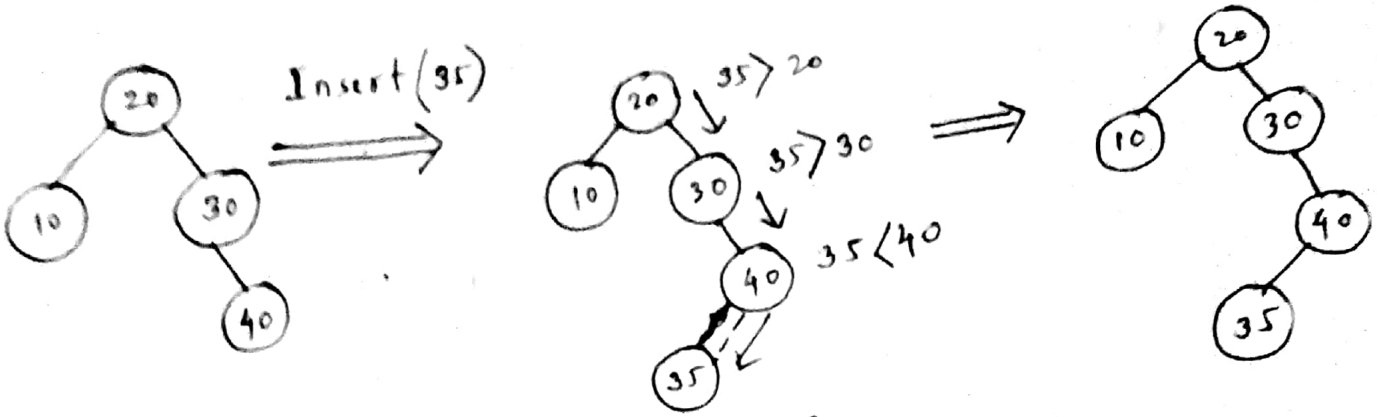


Completed Binary Tree:



## Insertion in a Binary Search Tree:




## Deletion from a Binary Search Tree: