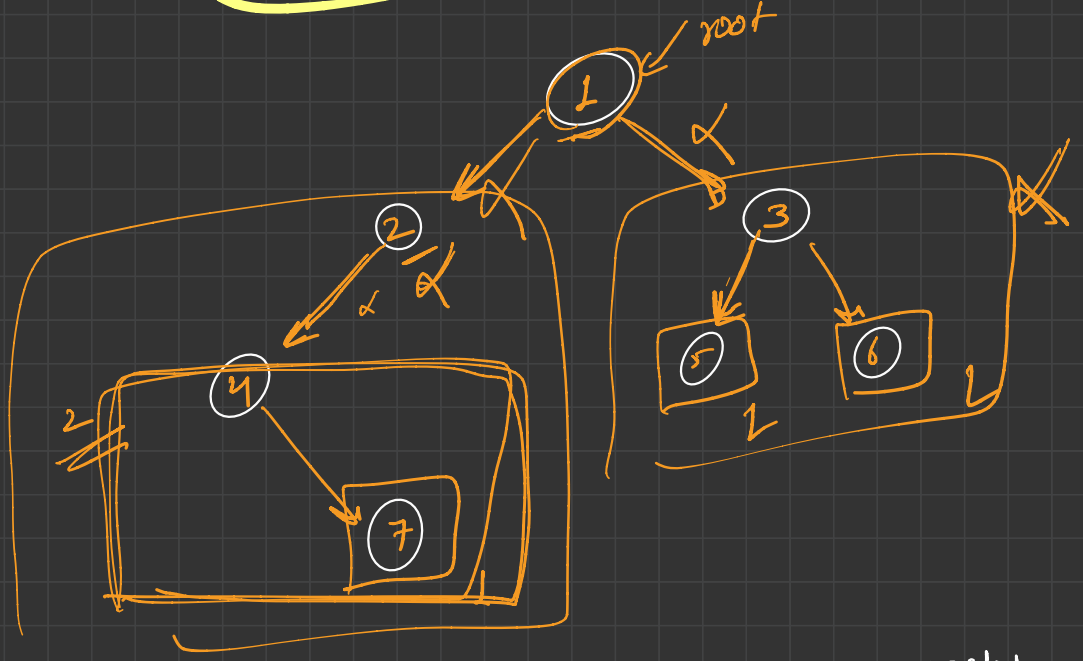
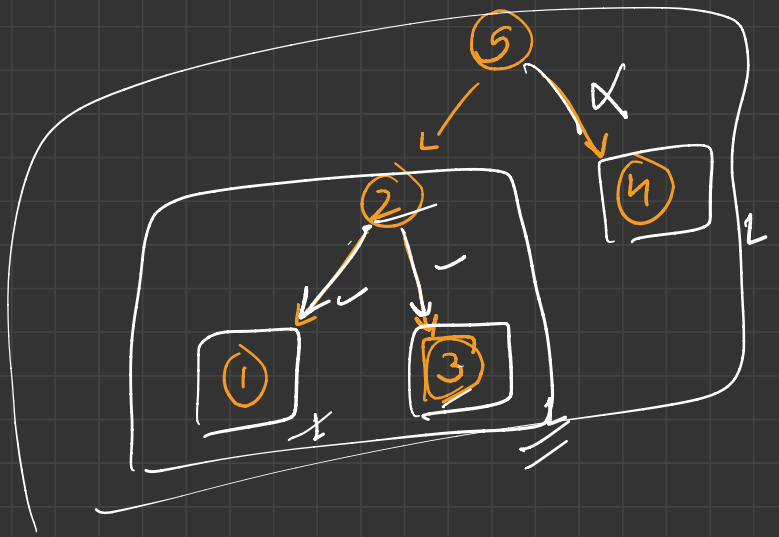
Largest BST



approach:-



L → 3

→ Approach #1 → every node

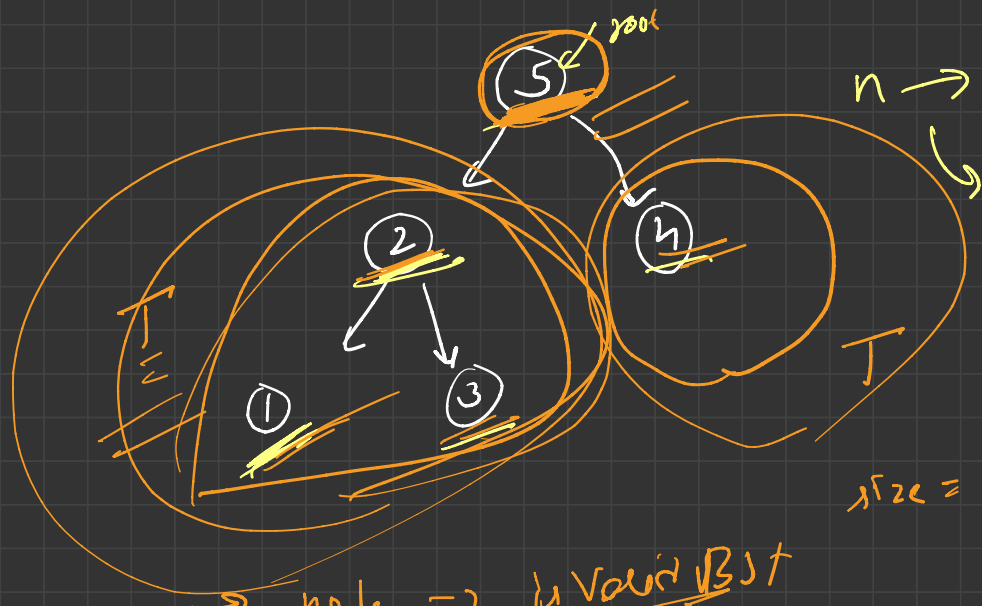
maxSize

Valid BST

size thar

Brute force

is BST → O(n)



n → is BST

↳ T.C → O(n²)

size = l.size + r.size + 1

→ node → is Valid BST

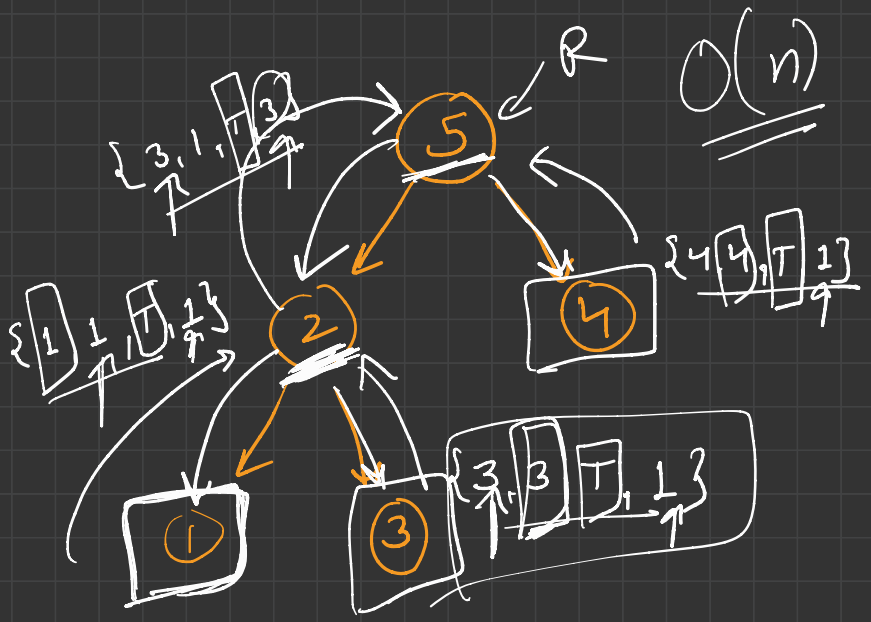
isBalanced BST

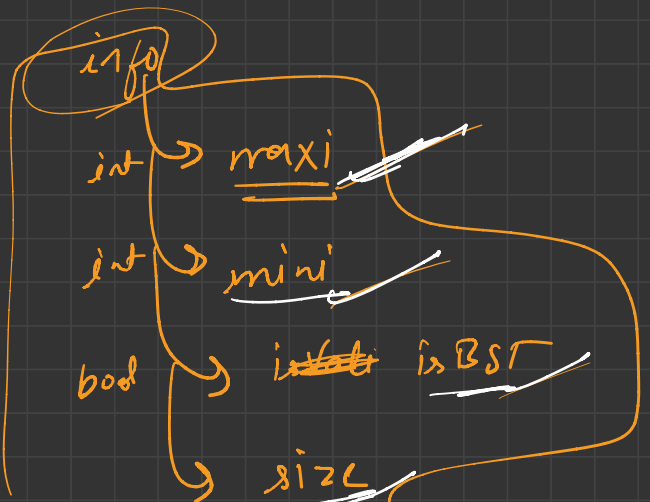
left subtree Valid BST → O(1)

right sub ——— → O(1)

(max.left) < root → data < right.(min)

→ Optimised Approach:





3 < 5 < 4 → Fa

1 < 2 < 3 → T

maxi = max(2, 3) = 3

mini = min(2, 4) = 1

valid = T

size = 1.size + ~~j.size~~ 5.size + 1

= 1 + 1 + 1 = 3

maxSize return

3

99.999 %

Dry Run → 3-4 examples

T.C → O(n)

H/W