unacademy

Trees - Part V

Course on Data Structure

Pankaj Sharma • Lesson 25 • Aug 5, 2024

unacademy

CS & IT Engineering

Data Structure

Tree

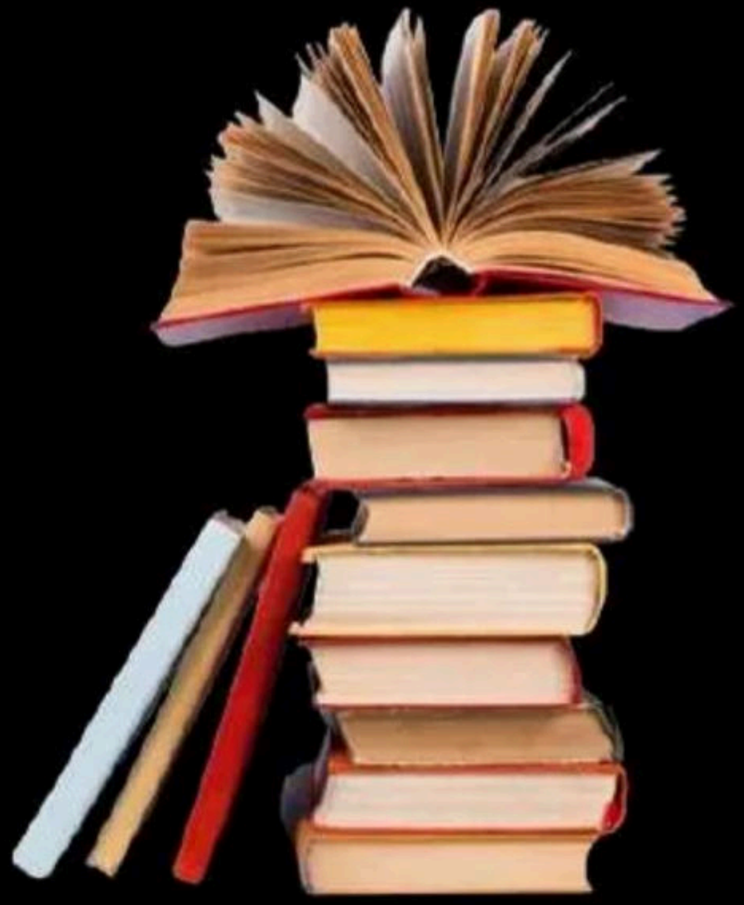
Lecture Number- 25

By- Pankaj Sir

unacademy

Topics

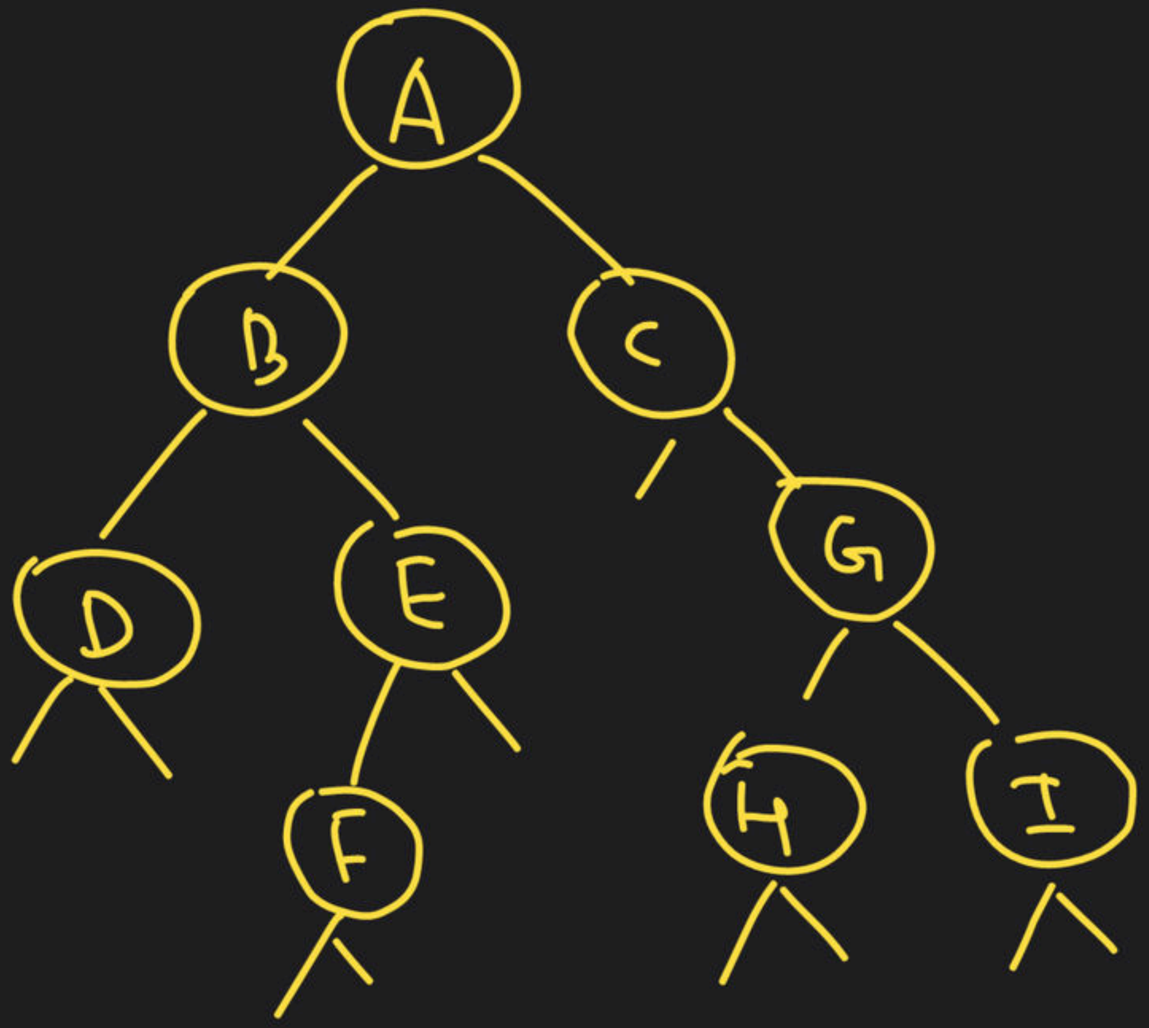
to be covered



1. Tree-IV

Pre: ABDEFCGHI

In: DBFEACHGI



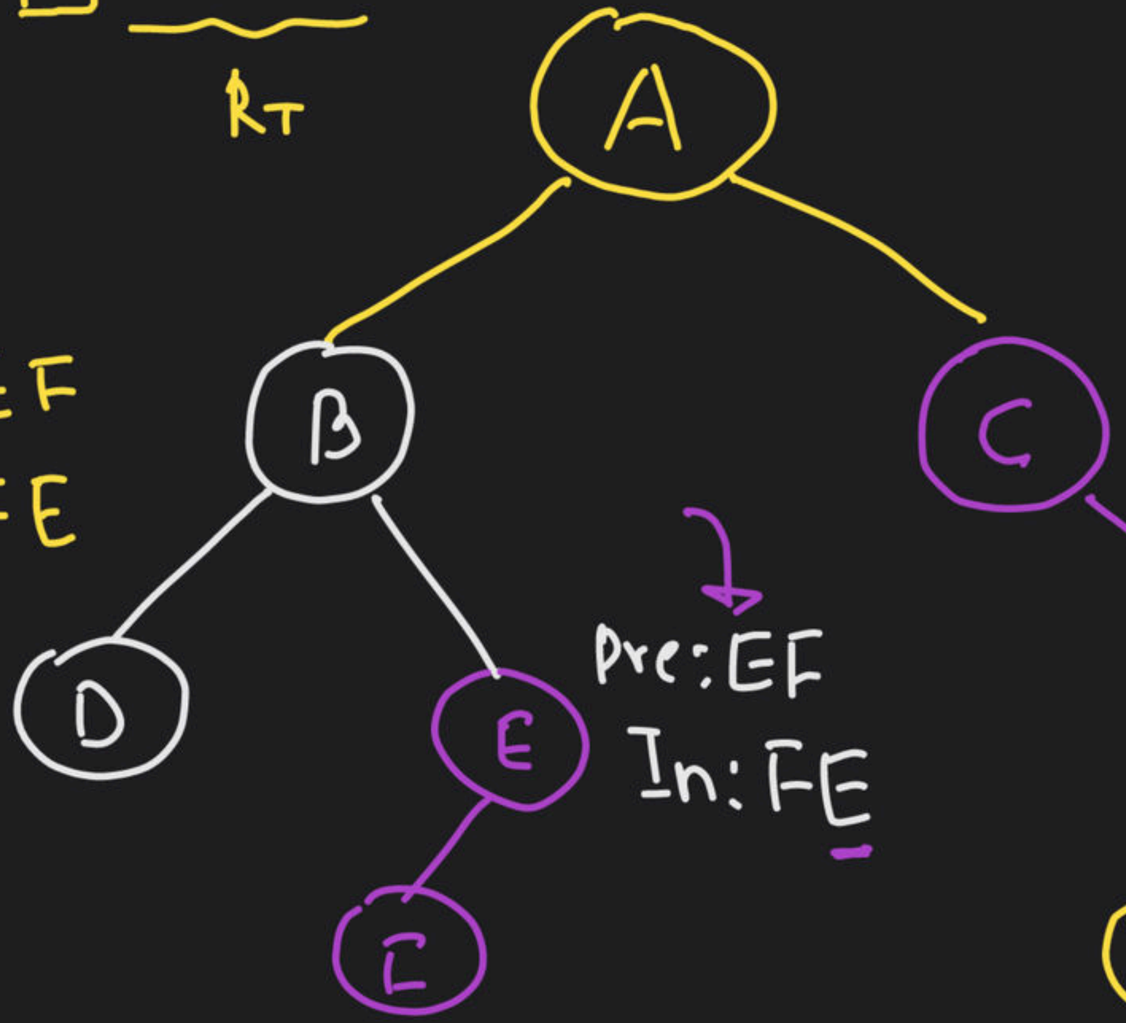
Pre: ABDEFCGHI

In: DBFEACHGI

LT RT

Root LT, RT

LT, Root, RT

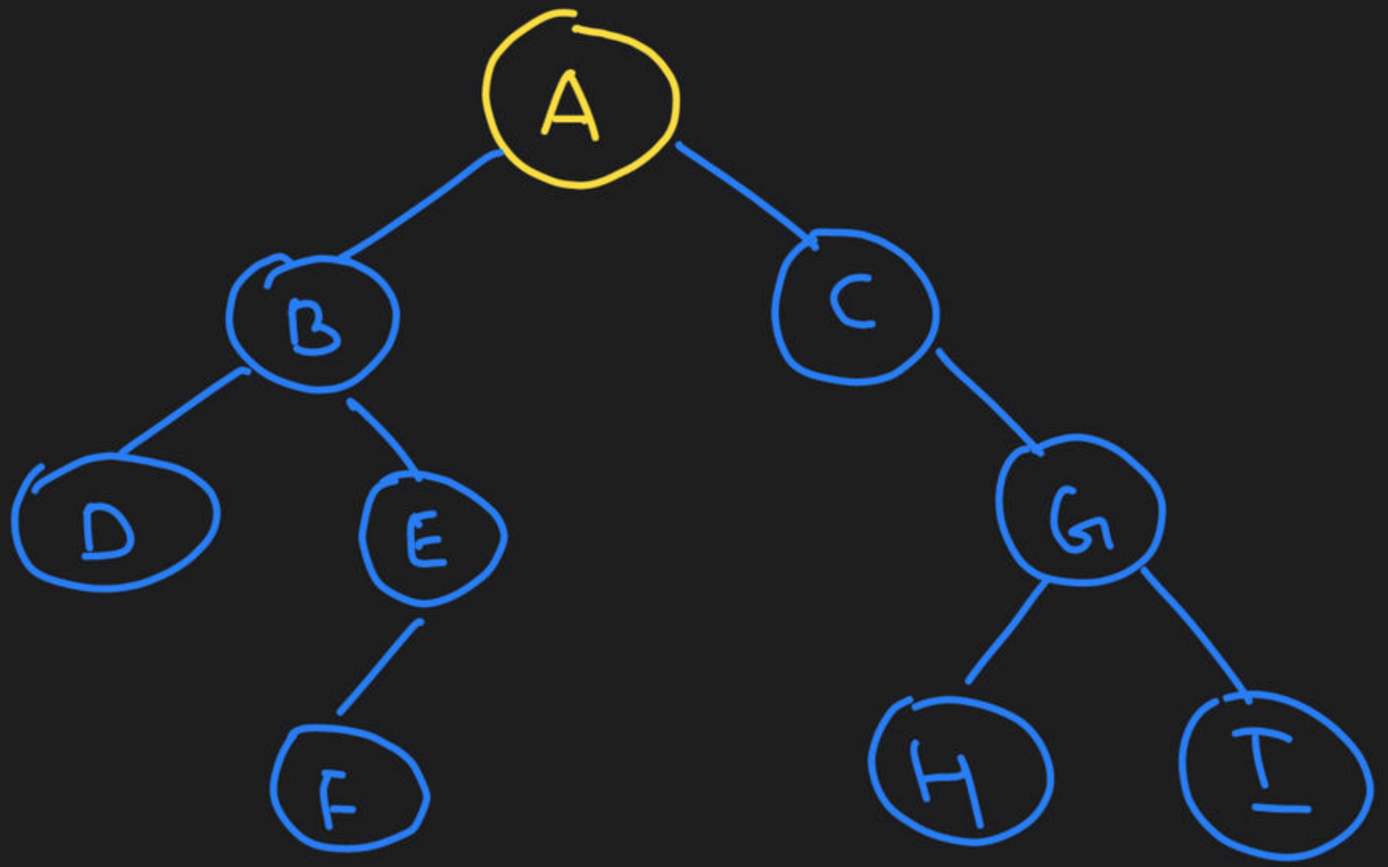


Pre: BDEF
In: DBFE

Pre: EF
In: FE

Pre: CGHI
In: CHGI

Pre: GHI
In: HGI



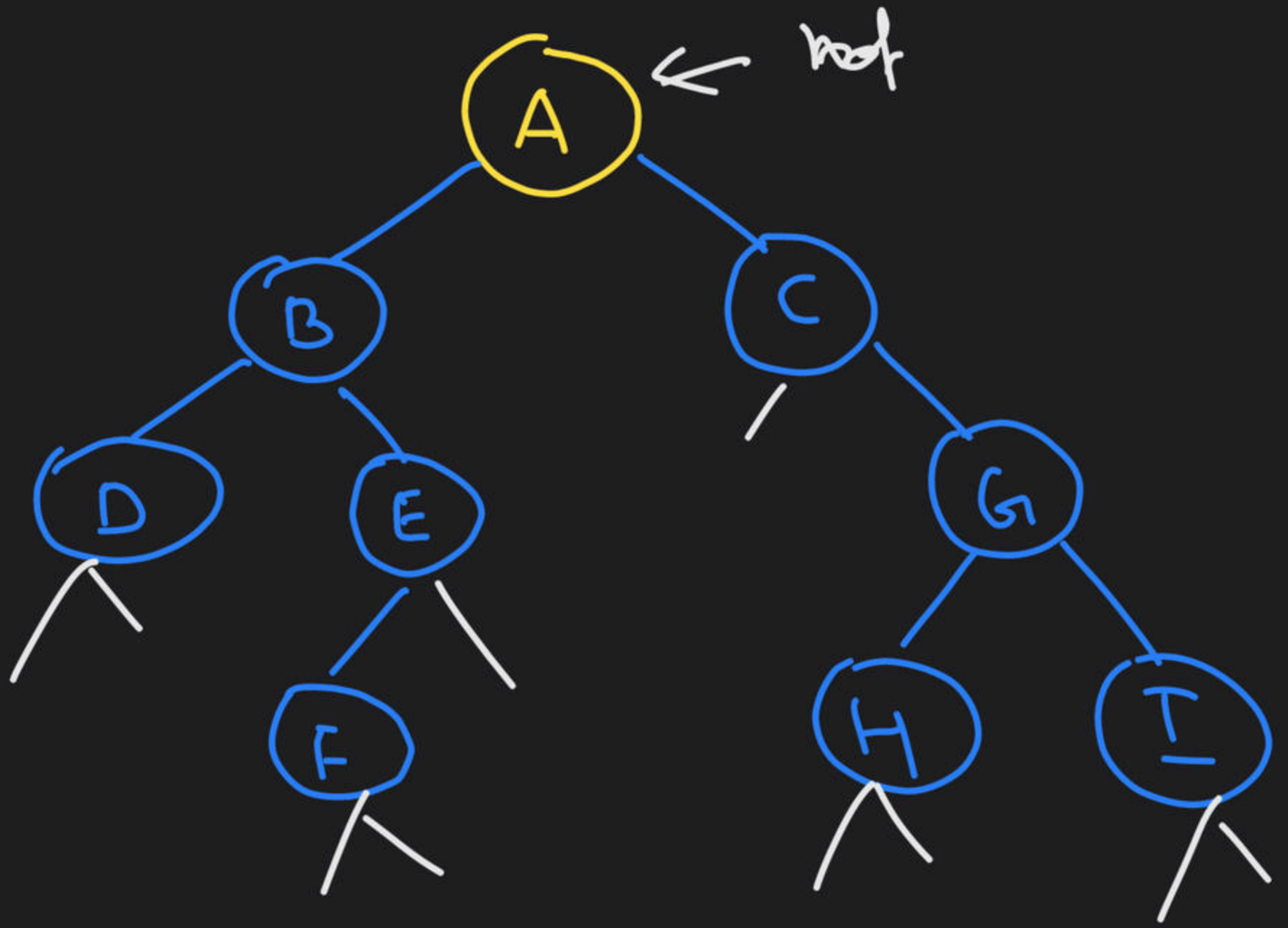
Pre : A B D E F C G H I

In : D B F E A C H G I

unacademy

Post : DFEBHIGCA

In: DBFEACHGI



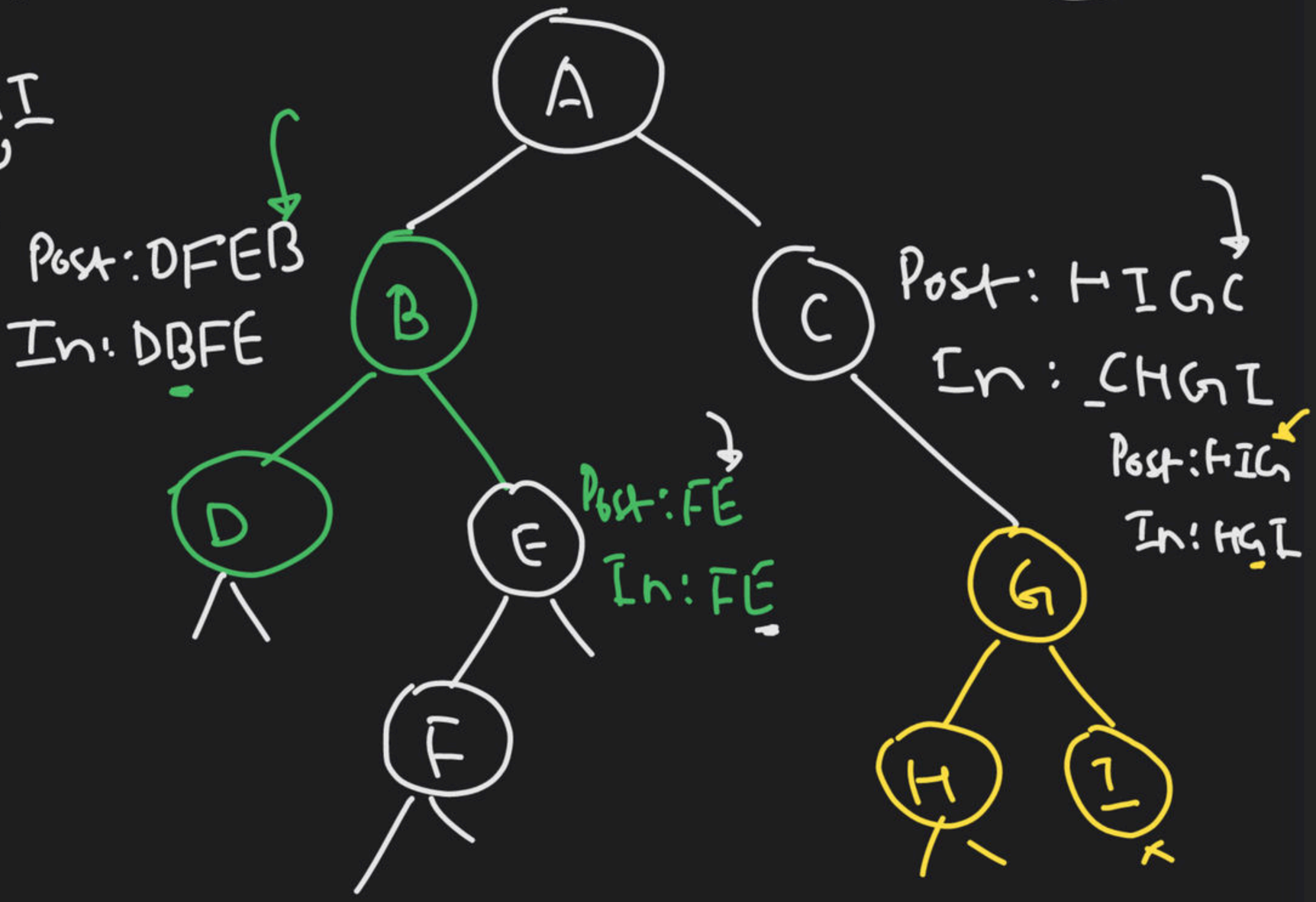
unacademy

Post : DFEBHIGCA

Post : LT, RT, Root

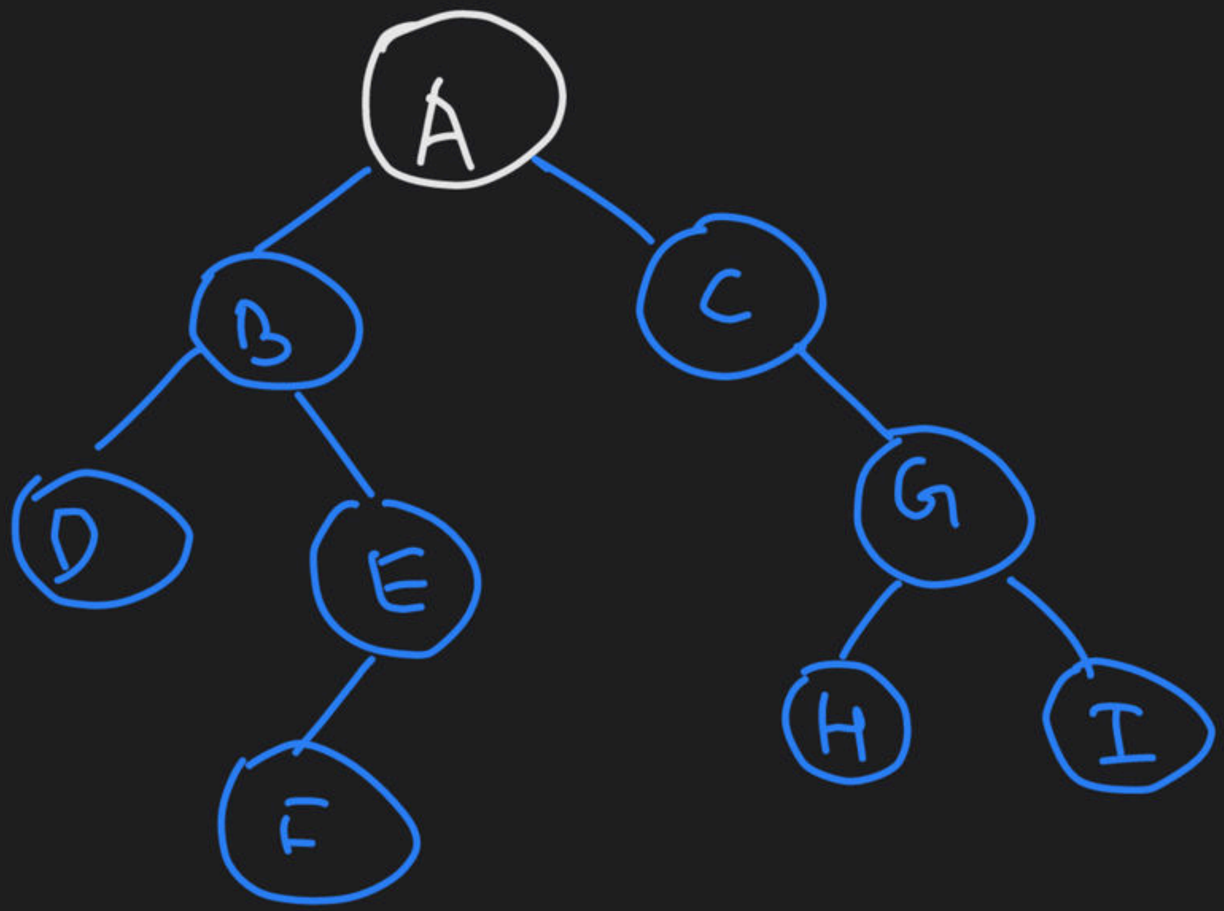
In: DBFEACHGI

LT RT

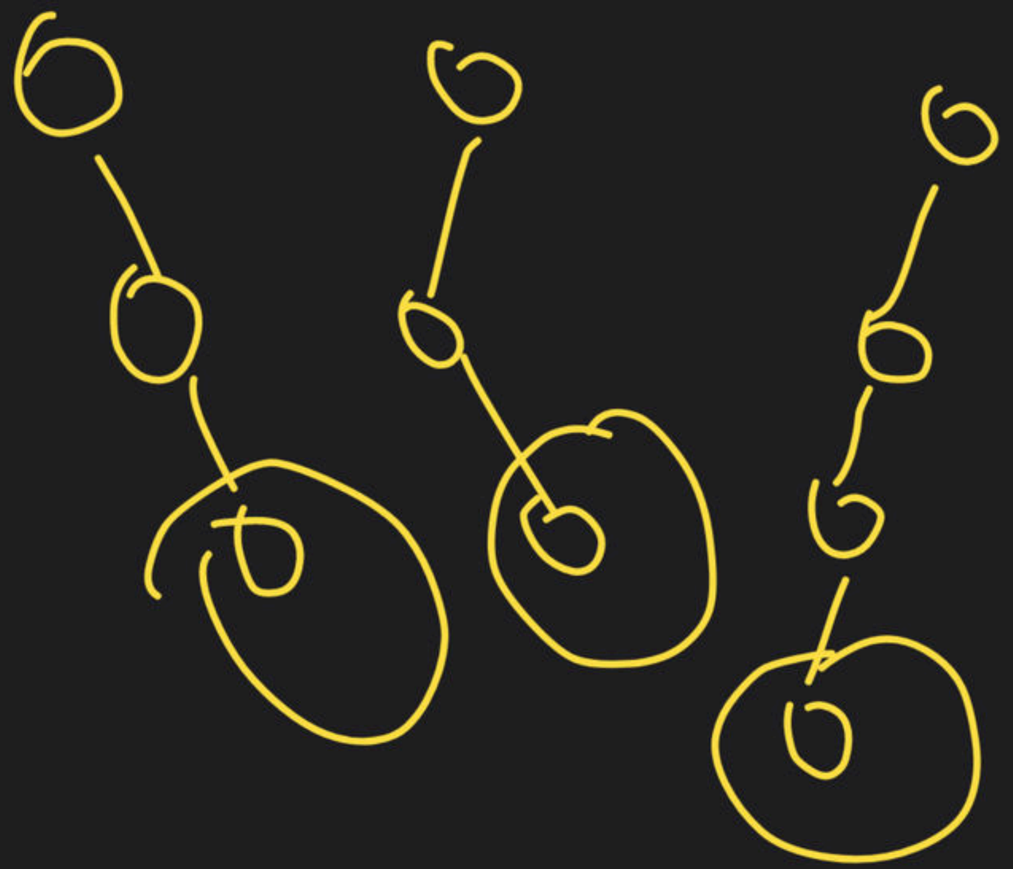


Post : D F E B H I G C A

In: D B F E A C H G I



Binary Tree with only 1 leaf node.



Every node has
0 or 1 child

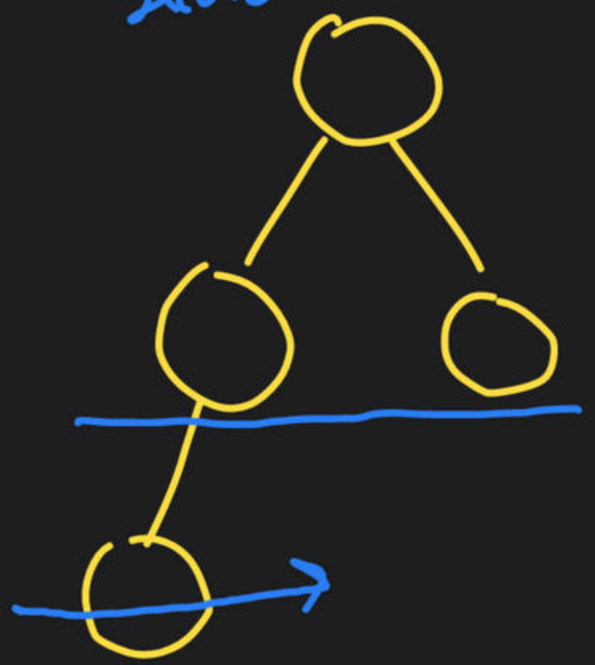
Q1. # binary trees are possible with $h=3$ & exactly 1 leaf node.

Q2. # binary trees with $h=7$ & exactly 1 leaf node.

Complete Binary Tree

A CBT is a binary tree which is full upto second last level and nodes at every level are filled from left to right in order.

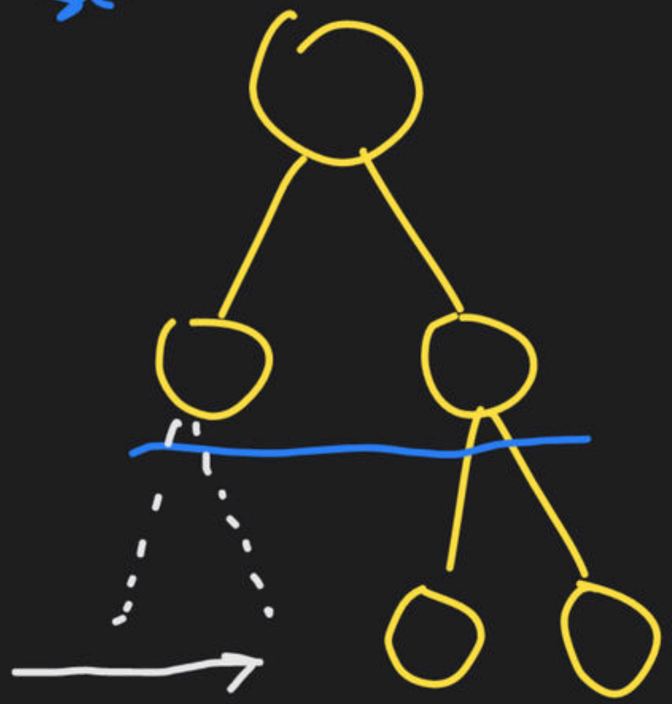
Second last ✓



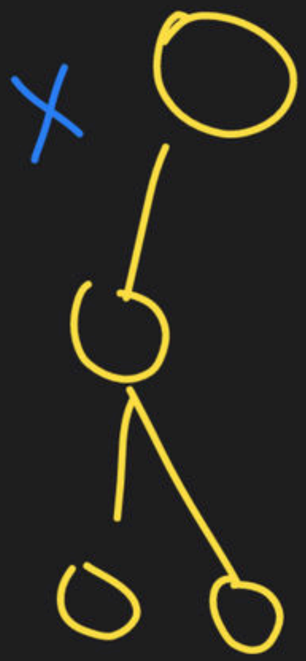
CBT

✓

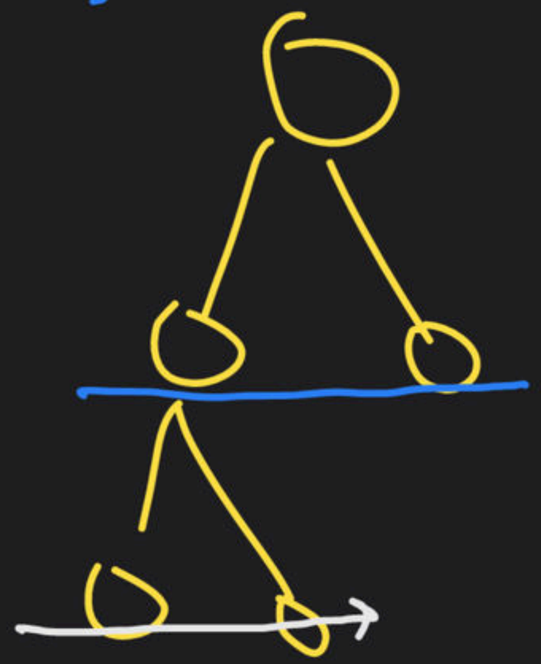
Second last



Not a CBT



Second last



h = 2

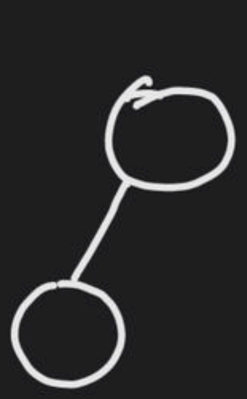
✓

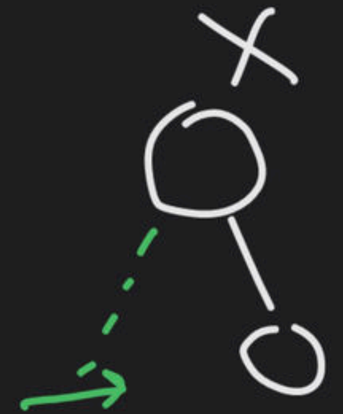
CBT

unacademy

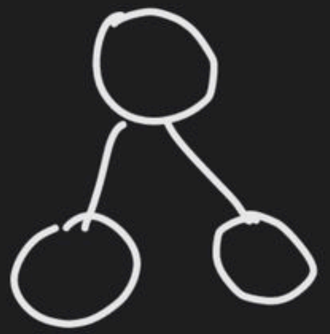
1. Structure of a CBT with 1 node

2. Structure of a CBT with 2 node

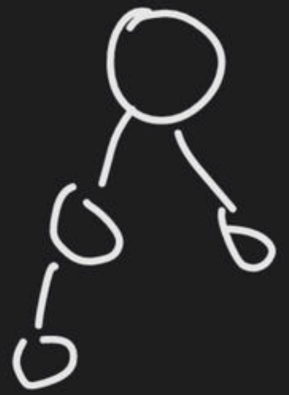




3. Structure of a CBT with 3 nodes



4. Structure of a CBT with 4 nodes

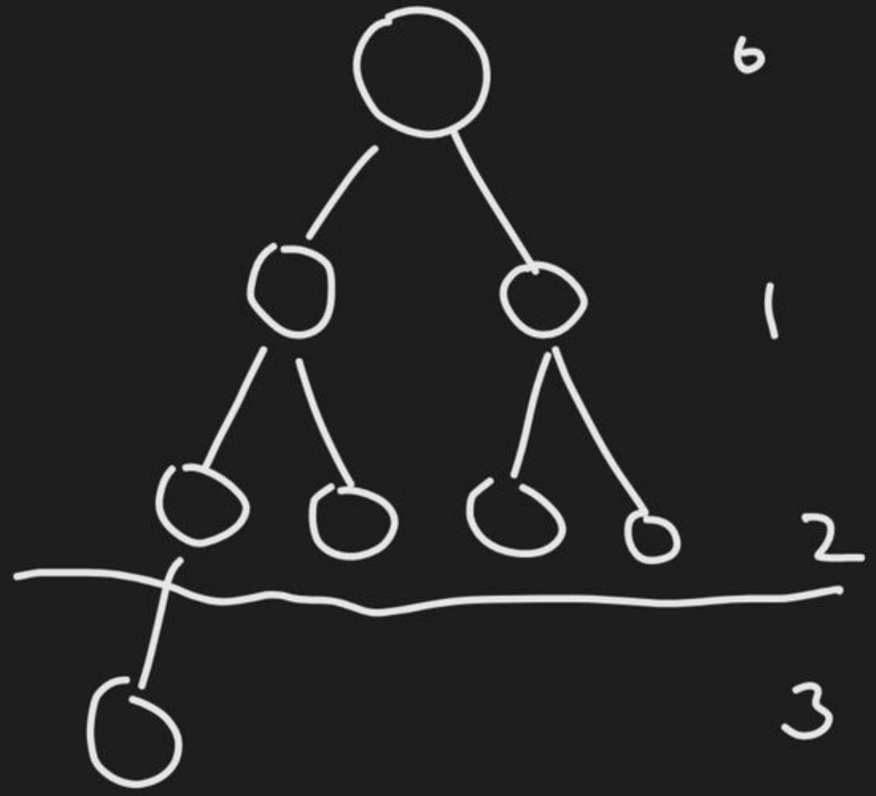


** The structure of a CBT with k nodes is always fixed (unique)

Max. no. of nodes in a CBT of h height

$$= 2^{h+1} - 1$$

Min. no. of nodes in a CBT of h height

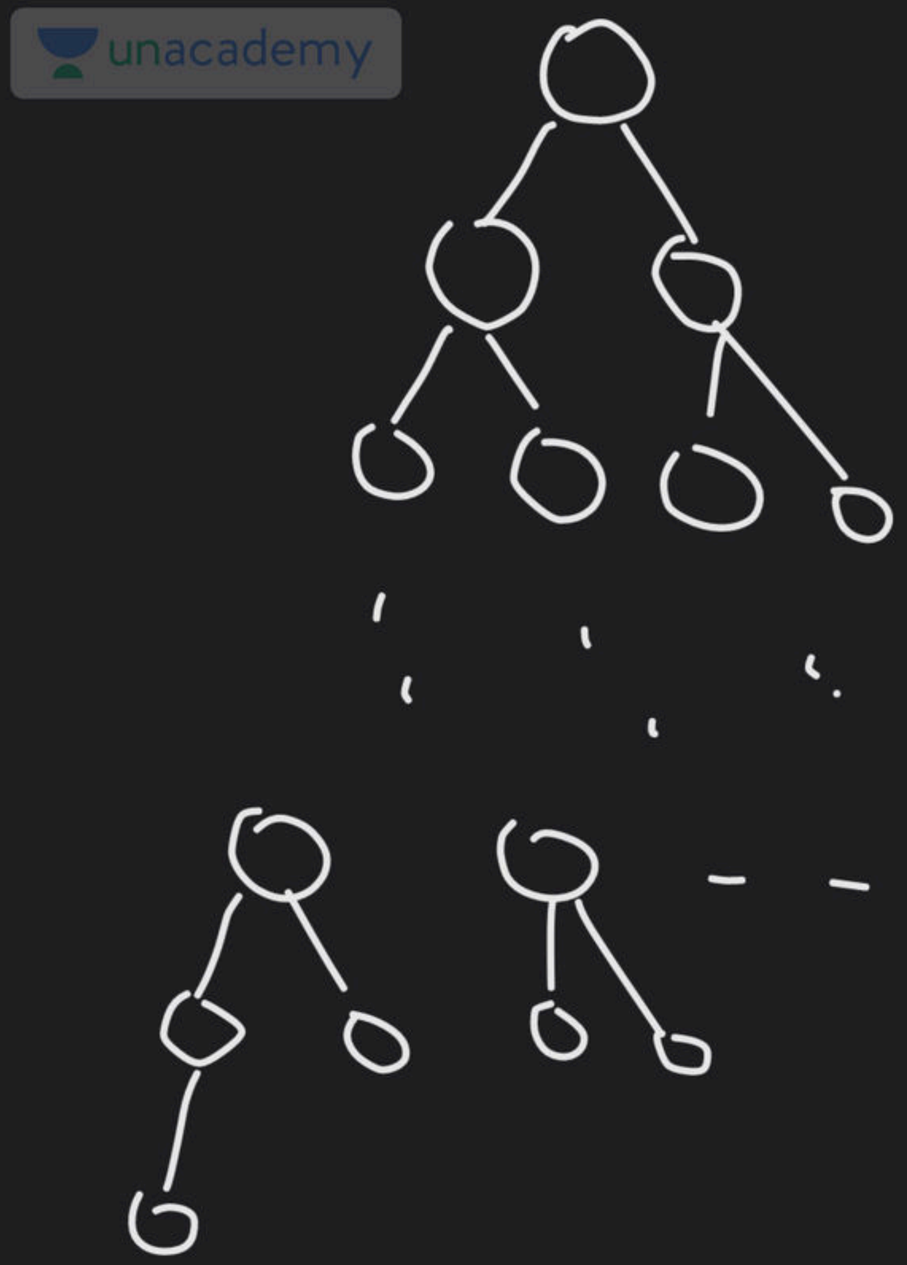


$h=3$

$=$

$=$

$N = (2^0 + 2^1 + 2^2) + 1 = (2^3 - 1) + 1 = 2^3$



level	# nodes
0	2^0
1	2^1
2	2^2
$h-2$	2^{h-2}
$h-1$	2^{h-1}
h	1

$$n_{min} = \left(2^0 + 2^1 + 2^2 + \cdots + 2^{h-1}\right) + 1$$

$$= \left(2^h - 1\right) + 1$$

$$= 2^h$$

$$n_{min} = 2^h$$

$$n_{max} = 2^{h+1} - 1$$

$$2^h \leq n \leq 2^{h+1} - 1$$

unacademy

$$2^h \le n \le 2^{h+1} - 1$$

$$2^h \le n$$

$$\log(2^h) \le \log n$$

$$h \log 2 \le \log n$$

$$h \le \frac{\log n}{\log 2}$$

$$\boxed{h \le \log_2 n}$$

$$n \le 2^{h+1} - 1$$

$$(n+1) \le 2^{h+1}$$

$$\log(n+1) \le (h+1)\log 2$$

$$\frac{\log(n+1)}{\log 2} \le h+1$$

$$h+1 \ge \log_2(n+1)$$

$$h \ge \log(n+1) - 1$$

unacademy

$n \leq 10$ && $n \geq 3$

$h \geq \log_2(n+1) - 1$

$3 \leq n \leq 10$

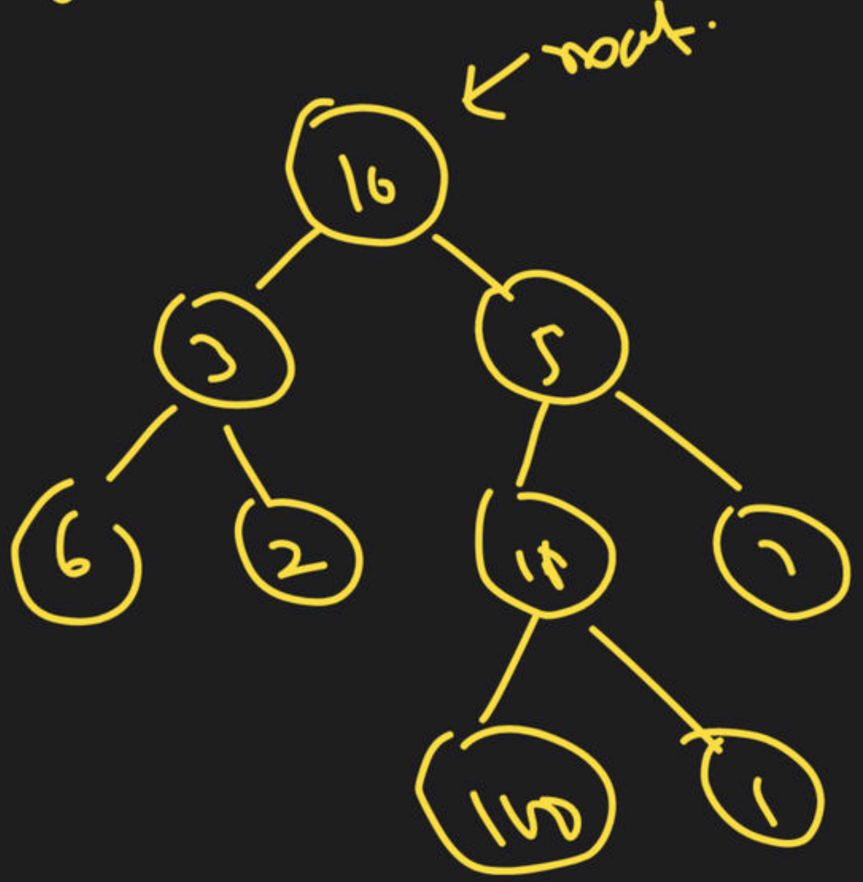
$h \leq \log_2(n)$

$$\log_2(n+1)-1 \leq h \leq \log_2(n)$$

unacademy

Binary Search Tree

① Why?



Given a key and a binary tree, find whether the key is present in the tree or not?

Traverse → O(h)

key = 99

unacademy

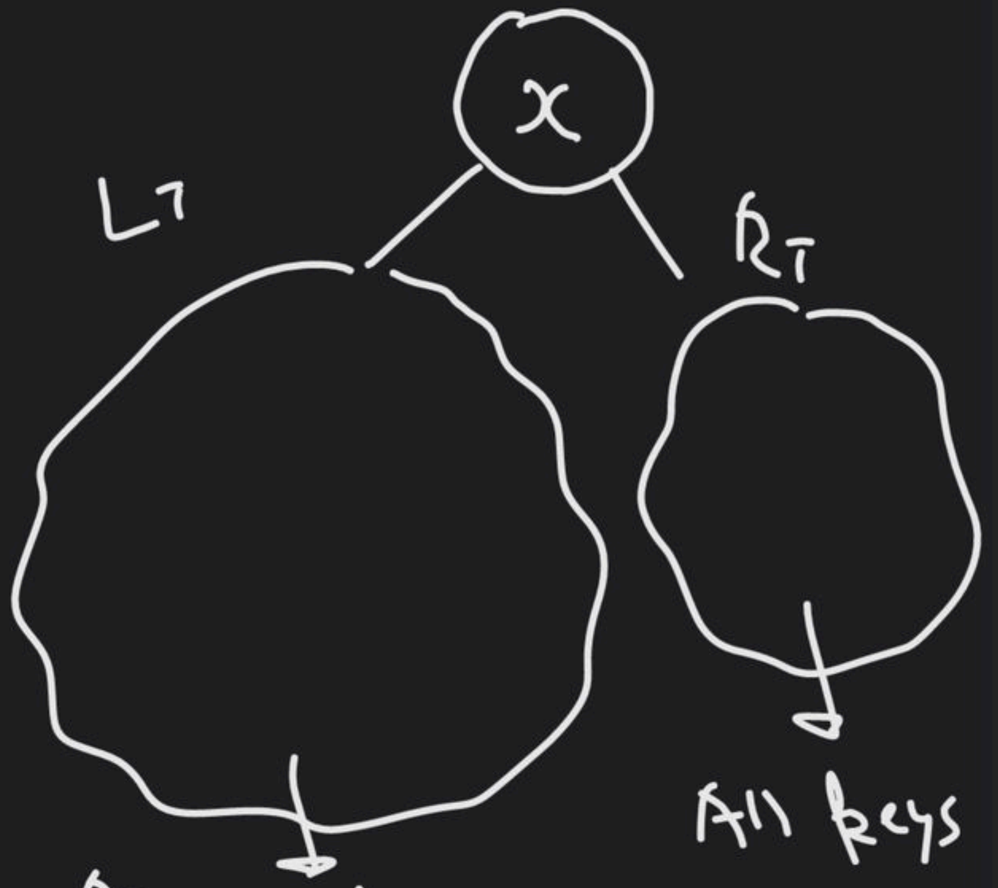
BST

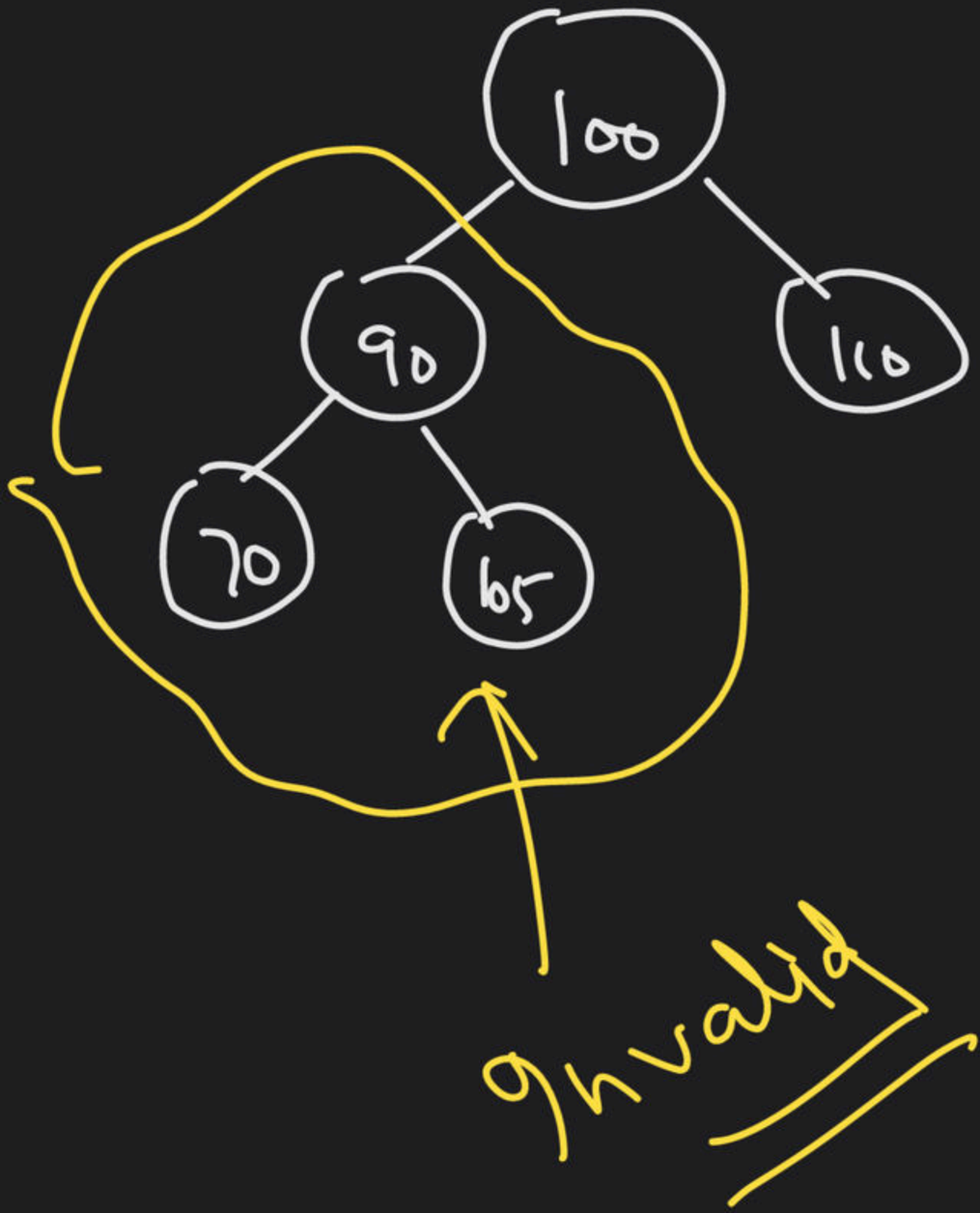
A binary search tree is a binary tree in which every node satisfies the following property:

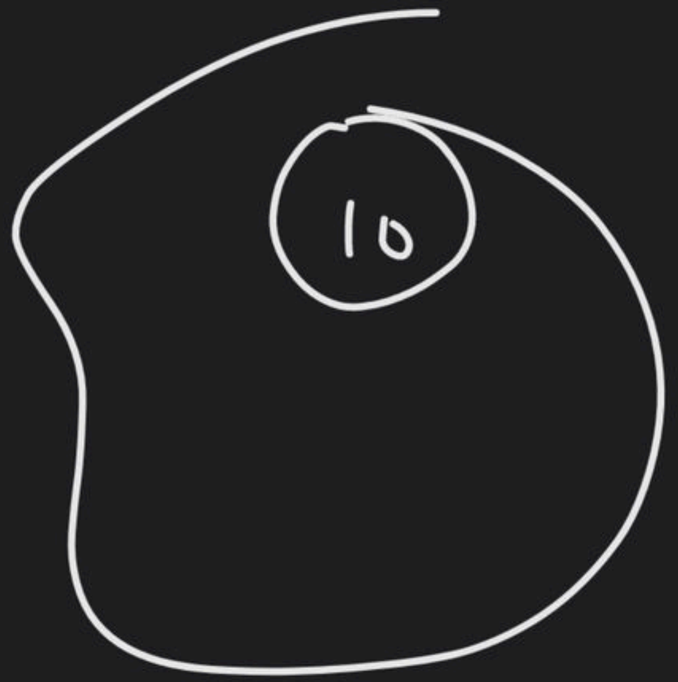
All the keys in the left subtree of a node are smaller than the key in the node.

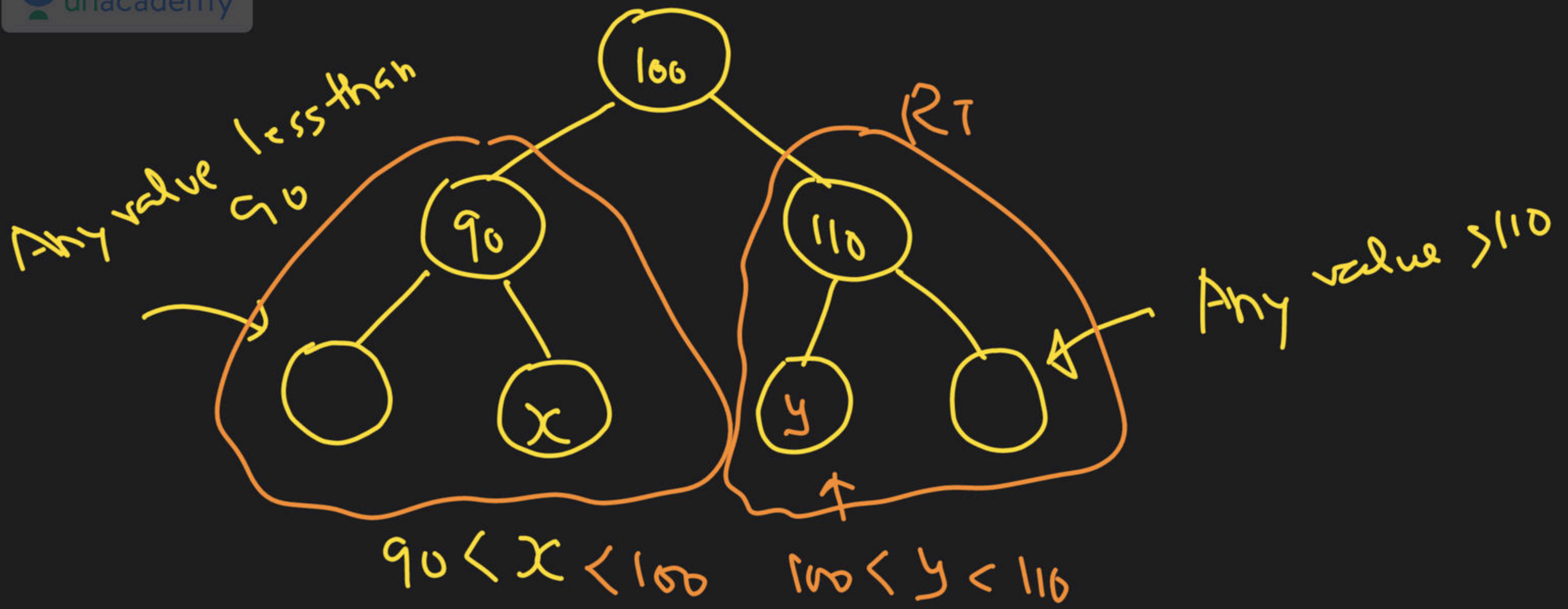
and

All keys in the right subtree of a node are greater than the key







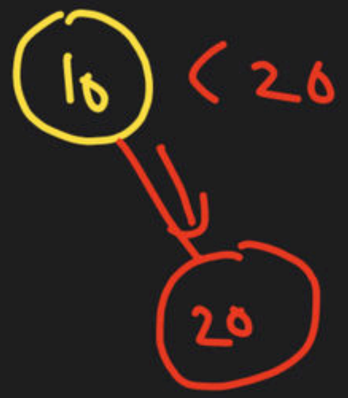

Any value lessthan 90
100
RT
90
110
x
y
Any value >110
90 < x < 100
100 < y < 110

unacademy

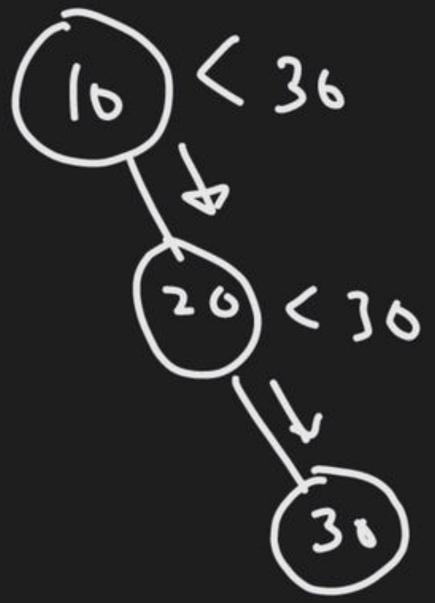
Q) Construct a BST by inserting keys 10, 20, 30 → in order into initially empty BST.

(i) Insert 10

(ii) Insert 20

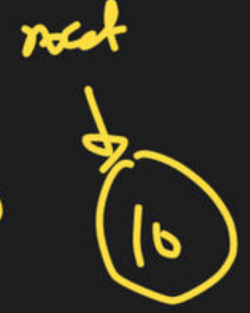


(iii) Insert 30



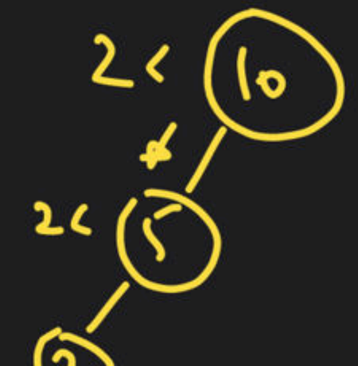
9) Construct a BST by inserting keys 10, 5, 2, 9, 3, 4 in order into an initially Empty BST.

(i) Insert 10

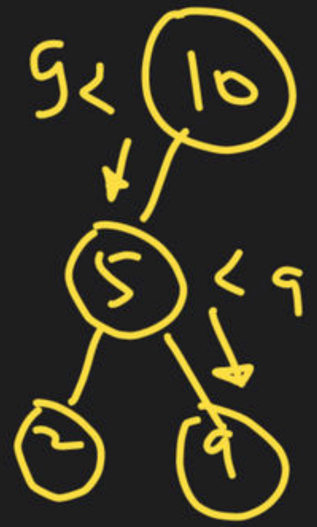


(ii) Insert 5

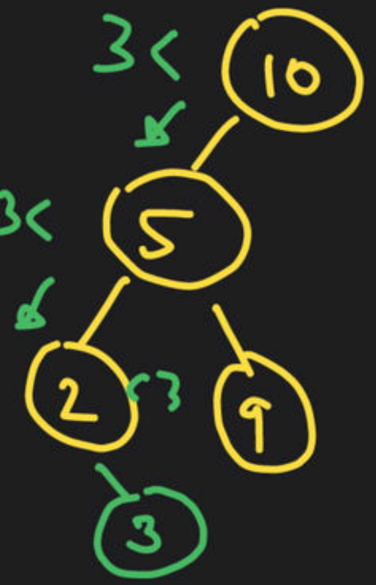
(iii) Insert 2



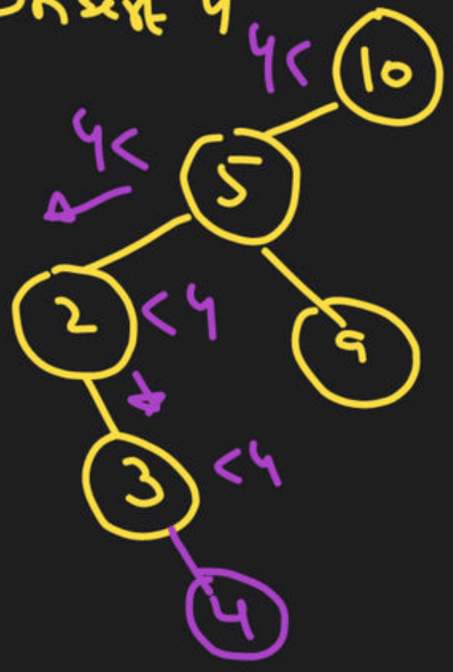
(iv) Insert 9



v) Insert 3



vi) Insert 4



No. of BST, when the insertion order of keys is fixed $\longrightarrow$ 1

Const. BST by inserting keys 1,2,3 ?

n = 3 keys → 5 BSTs (in any order)

order coulabe

a) 1, 2, 3 →

b) 1, 3, 2

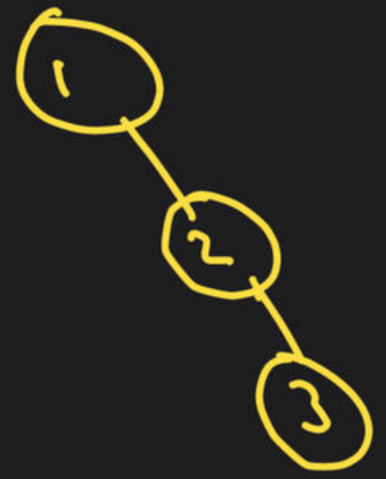
c) 2, 1, 3

d) 2, 3, 1

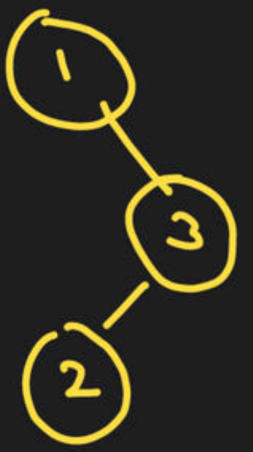
e) 3, 1, 2

f) 3, 2, 1

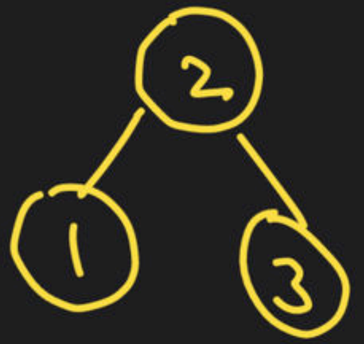
a)



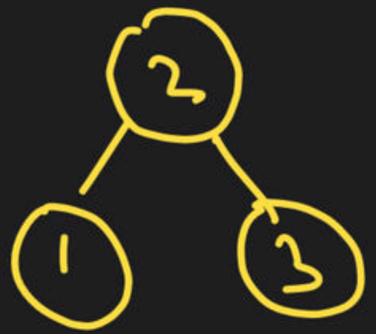
b)



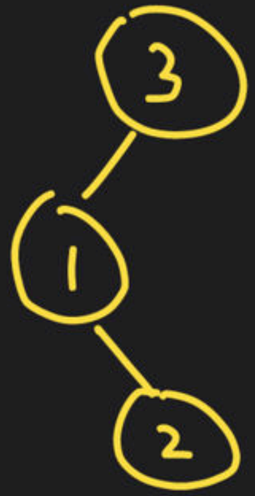
c)



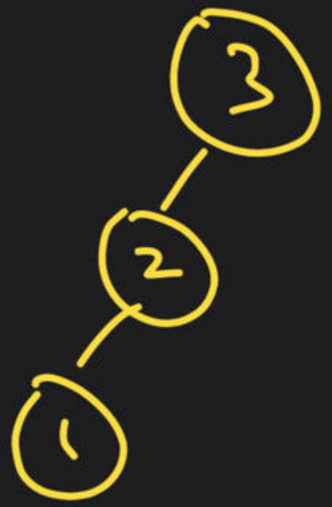
d)



e)



f)



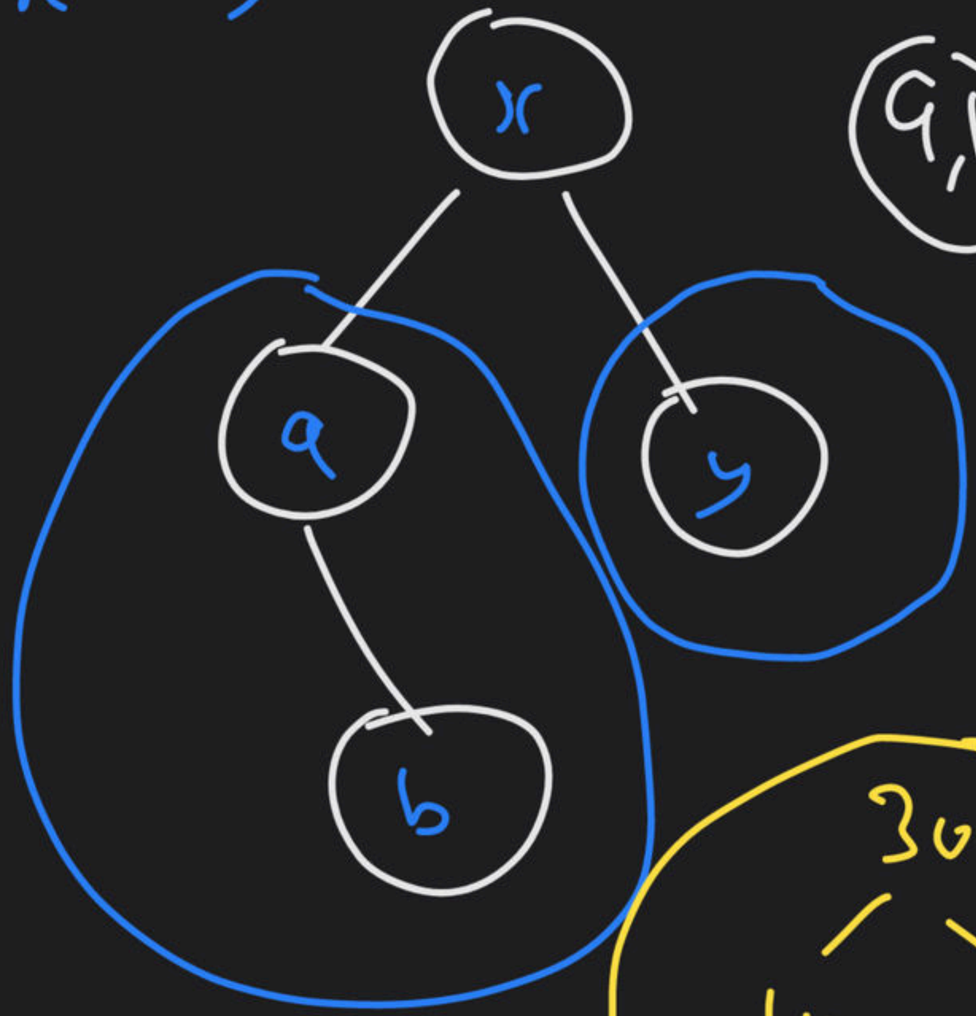
unacademy

$$\#\text{BSTs with } n \text{ keys} = \frac{2nC_n}{n+1}$$

Q Given a binary tree structure with n nodes & also n distinct keys are given.

In how many ways we can fill the structure so that it becomes a BST.

$a, b < x < y$

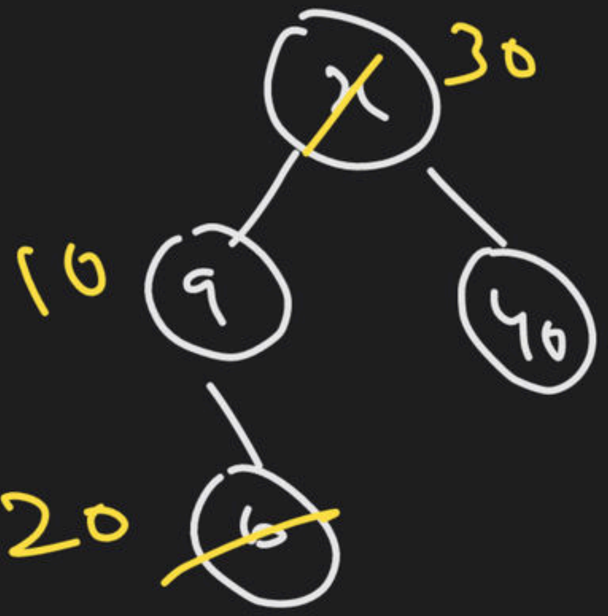


$a, b < x$

$h = y$

10, 20, 30, 40

y



10 20 30

10 20

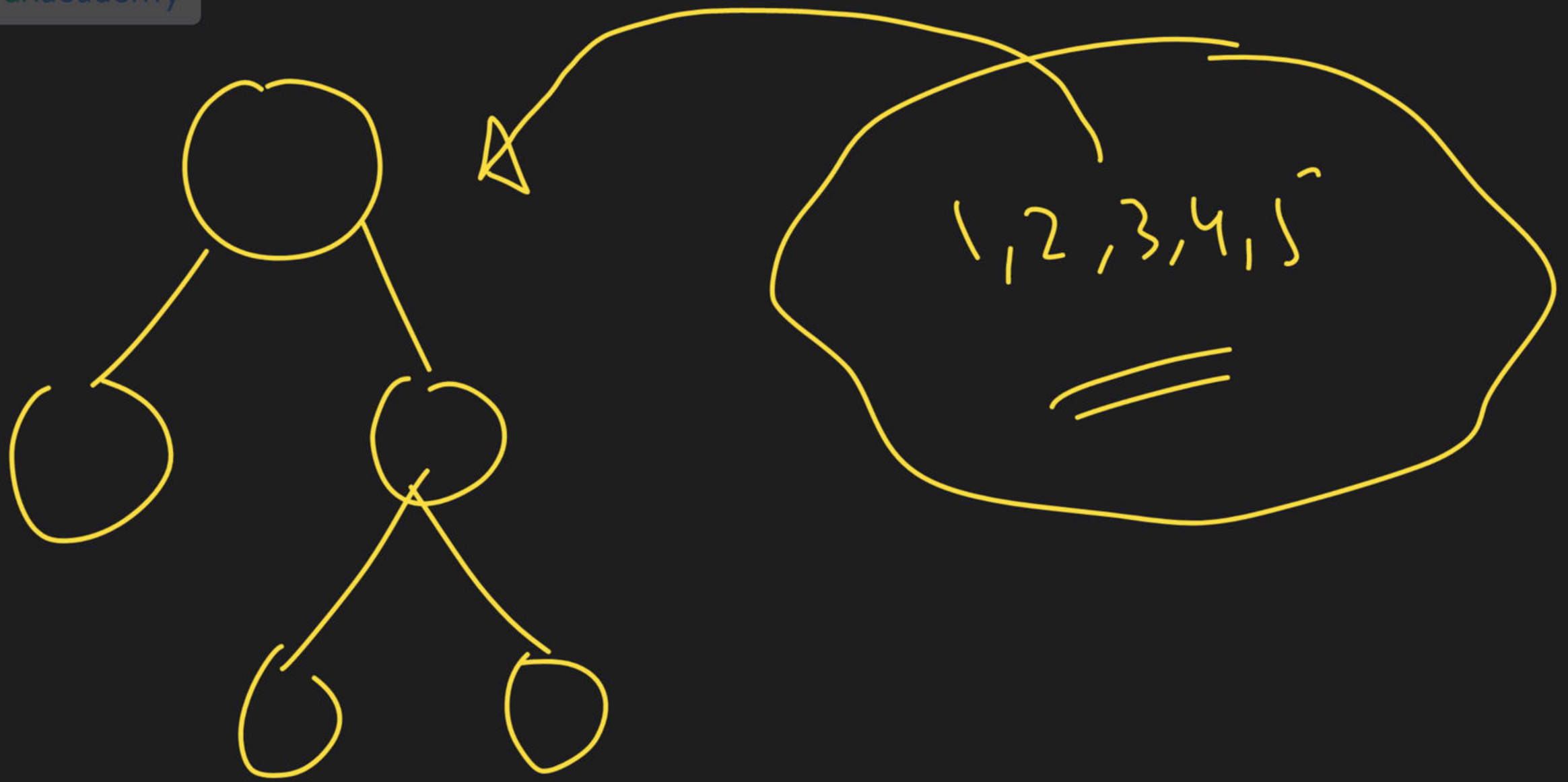
$b > a$

unacademy

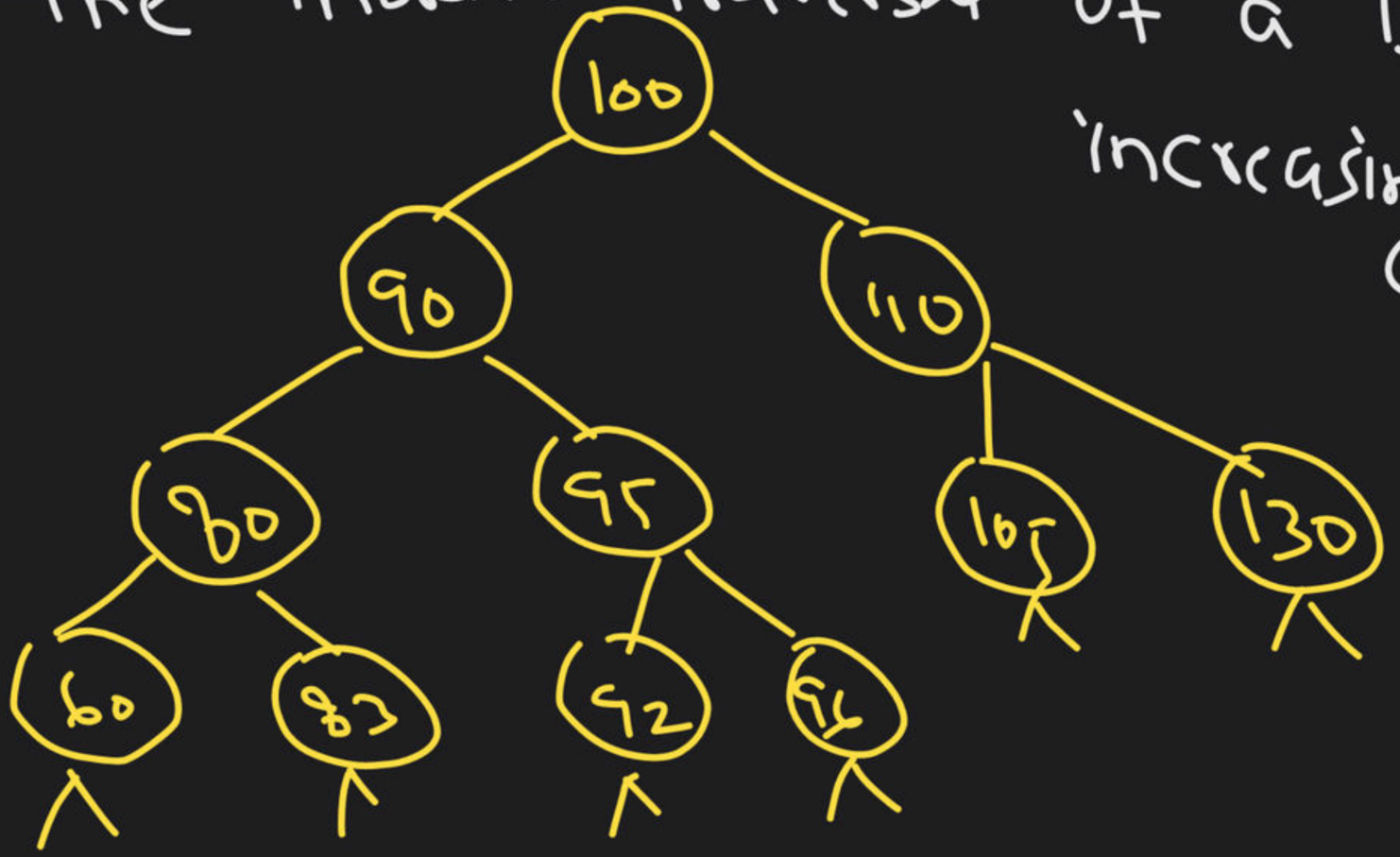
# of BSTs with n keys

$$= \left(\begin{array}{c}\text{structure with} \\ n \text{ nodes}\end{array}\right) \times \frac{1}{n+1}$$

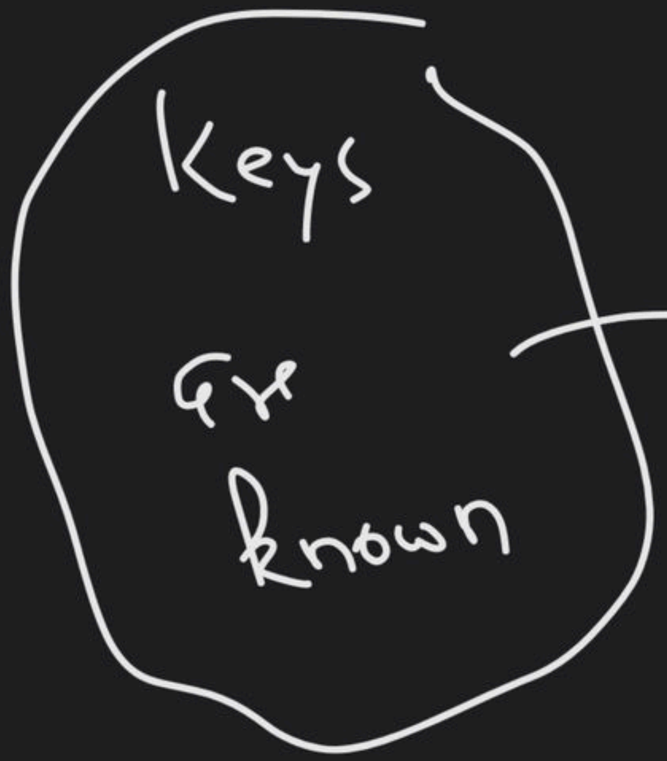
$$= \frac{2n C_n}{n+1}$$



The inorder traversal of a BST is always increasing order of keys.



In: 60, 80, 83, 90, 92, 95, 98, 100, 105, 110, 130



Inorder is known ?

BST → Preorder is given → keys → Asc. order

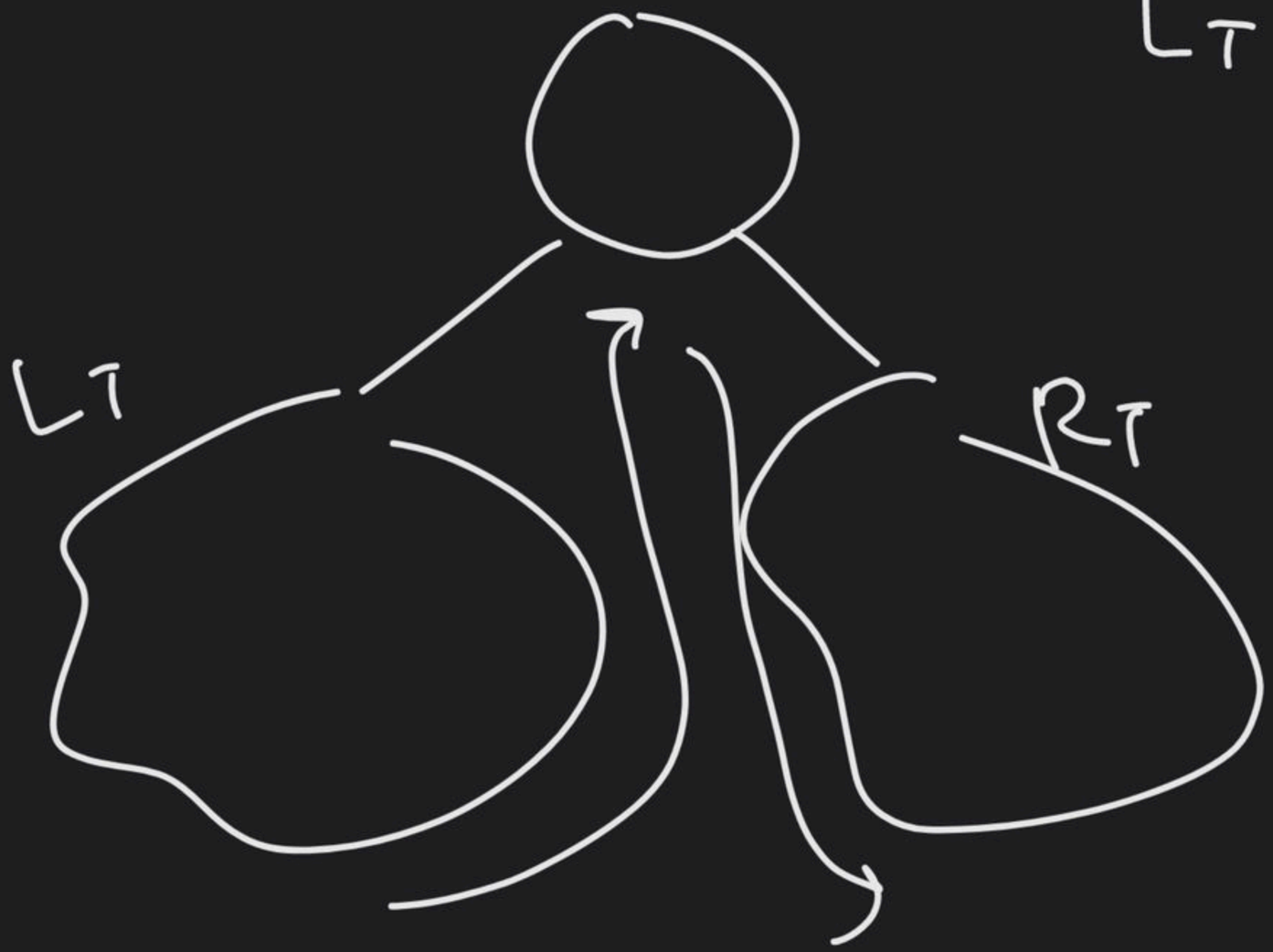
In.

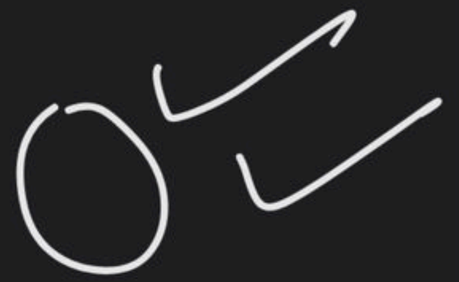
→ Find the postorder.

BST → Postorder is given

→ Find the preorder.

L_T, Roat, R_T





Doubt 3

unacademy

THANK YOU!

Here's to a cracking journey ahead!