CS & IT ENGINEERING

Data Structure

Tree
Chapter- 5
Lec- 10

By- Pankaj Sharma sir

TOPICS TO BE COVERED
Tree-X

infix : a x b + e/f - g

x /
+ -

Expression tree :

leaf : Operands

Internal nodes : operator

$\Rightarrow a \times b$

Inorder : a x b

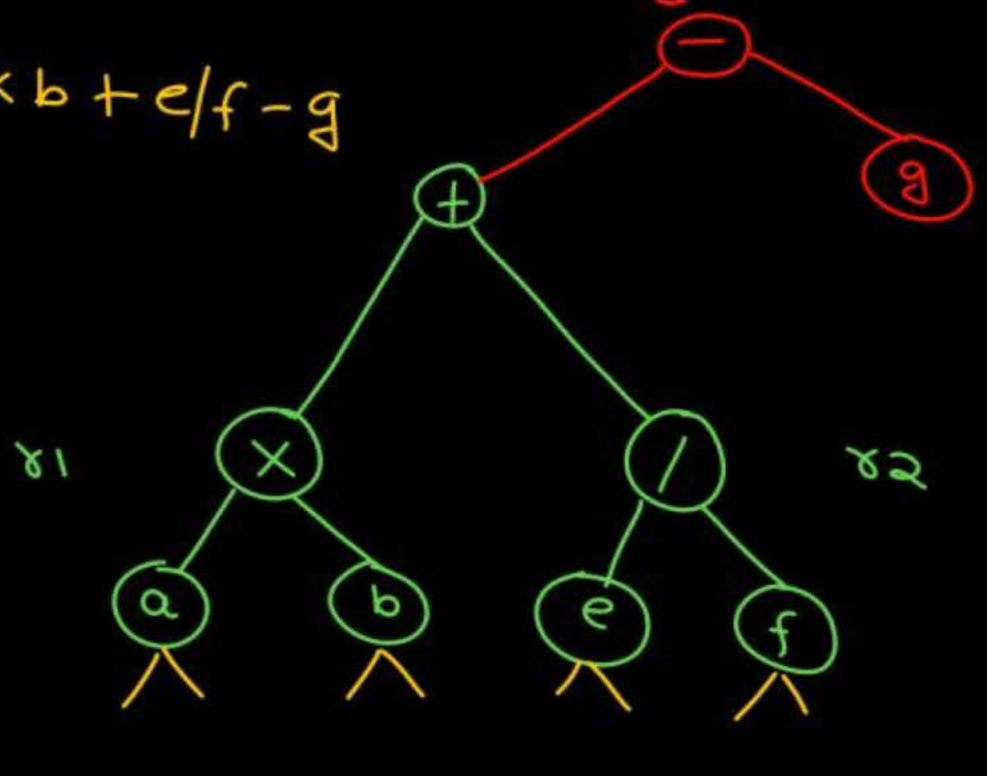
infix : (a x b) + e/f - g

γ1 γ2

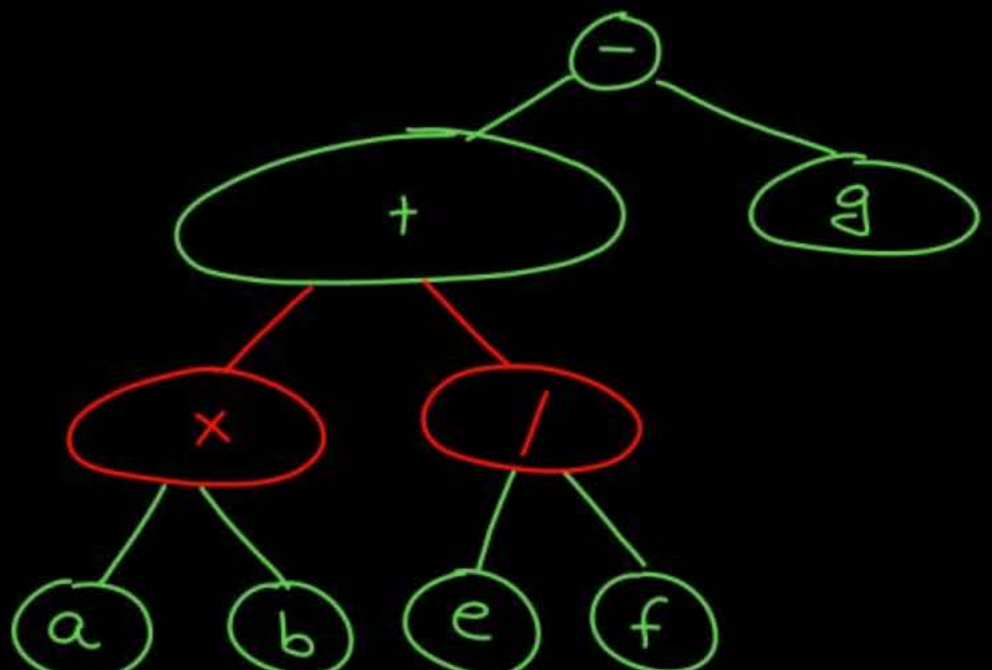
γ1 + γ2 - g

γ - g

inorder : a x b + e/f - g

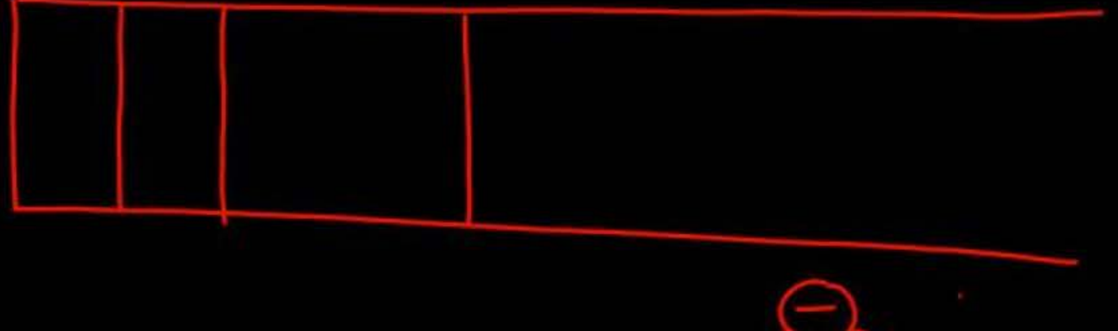


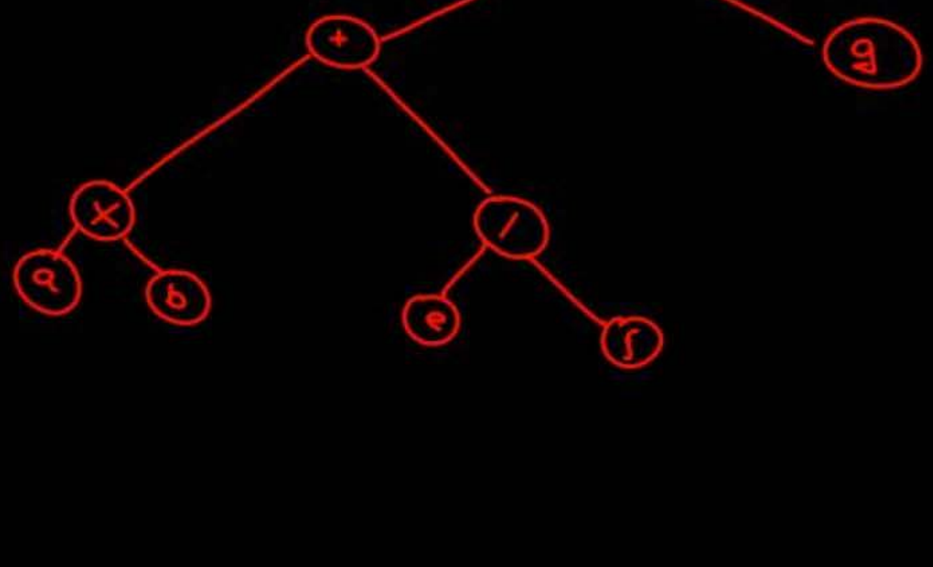
a x b + e / f - g ← last

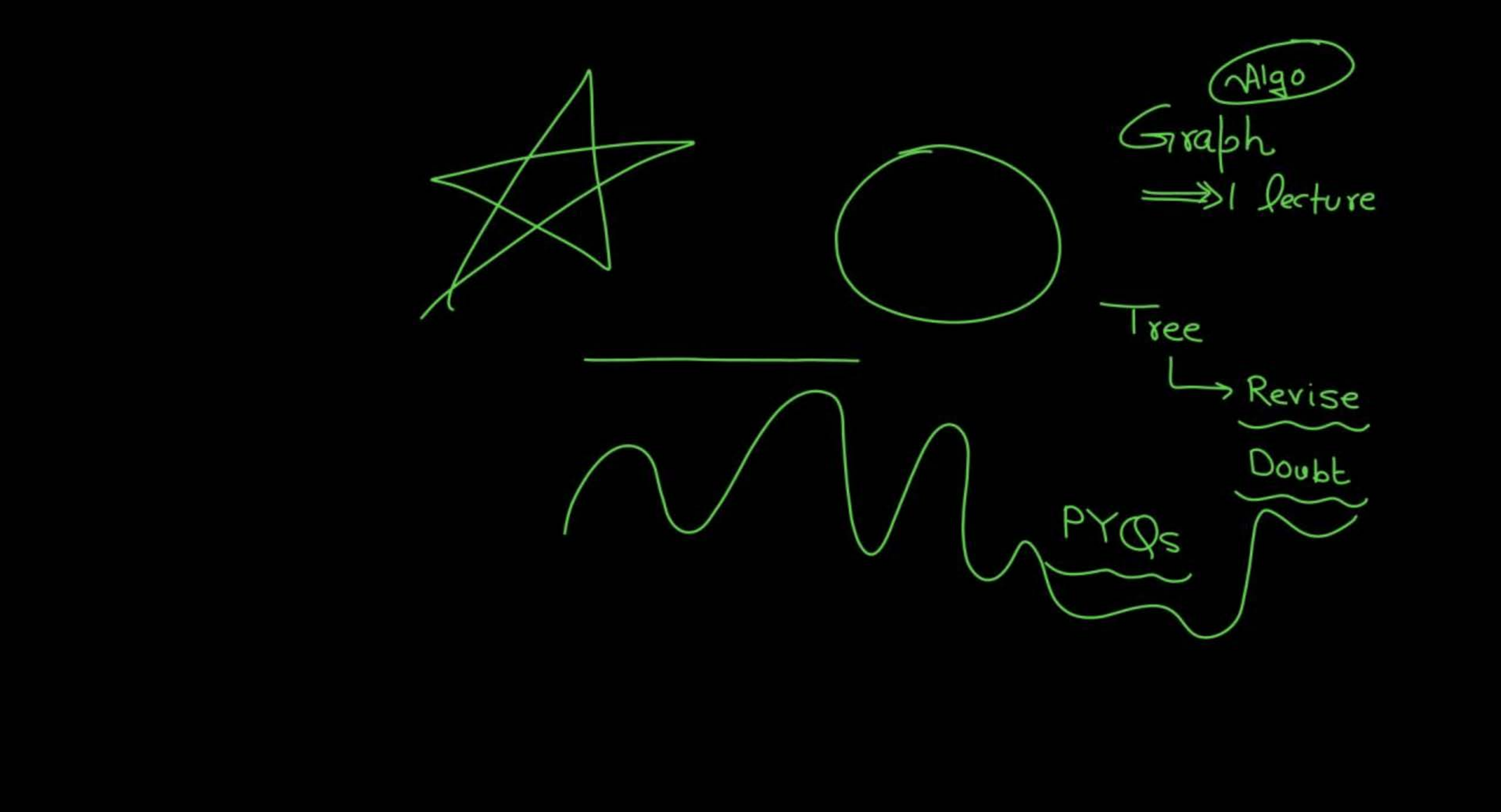


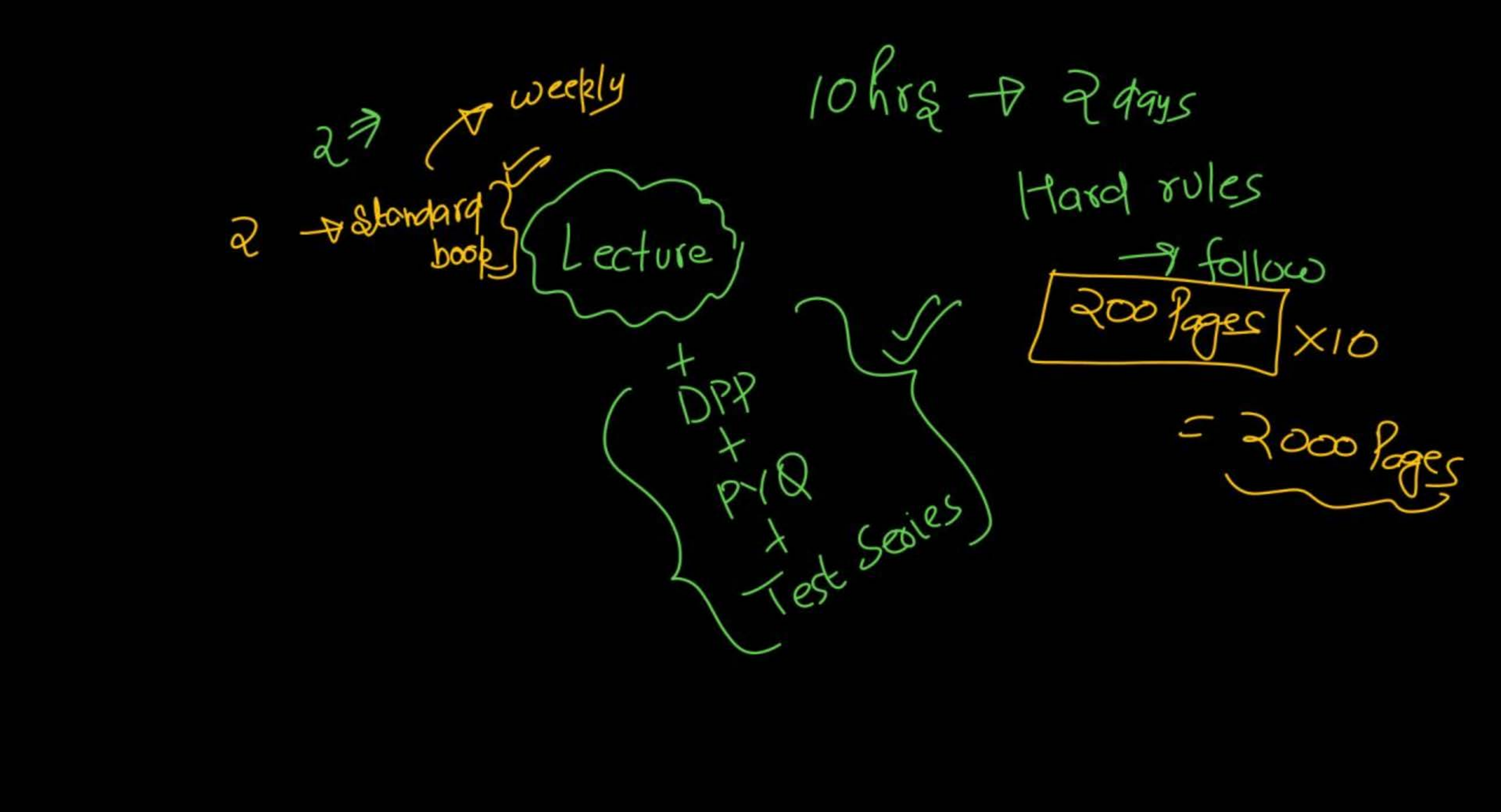
infix: a x b + e / f - g

Postfix: a b x e f / + g -






Algo
Graph
⟹ 1 lecture
Tree
Revise
Doubt
PYQs


2 ⇒ weekly
2 → standard book
Lecture
+ DPP + PYQ + Test Series
10 hrs → 2 days
Hard rules
→ follow
200 Pages ×10
= 2000 Pages

Infix : 2+3X4

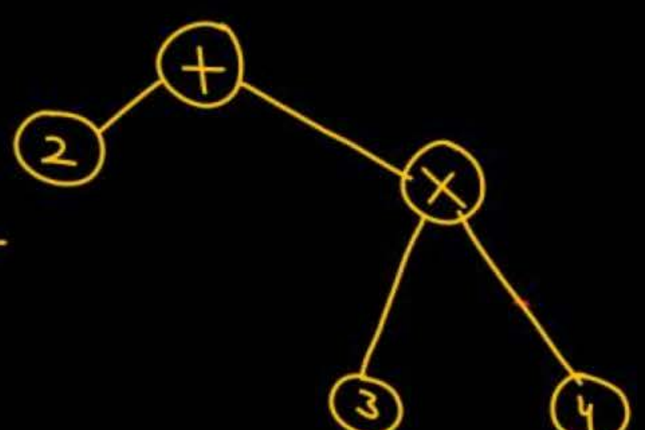
Prefix : +2X34

Reverse : 43X2+

inorder : 2+3X4

Preorder : +2X34

Postorder : 234X+

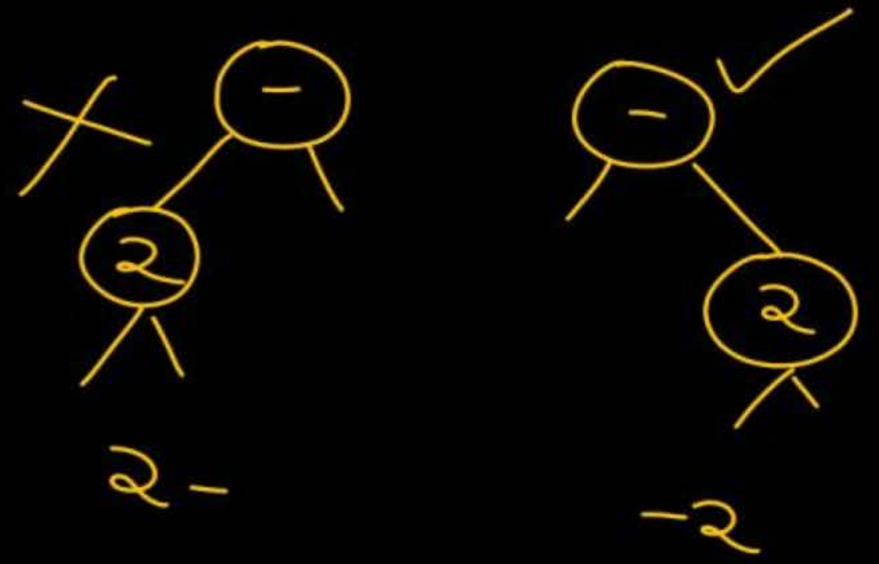


Unary operator

-2

Fail

exp: -2



infix: $3 \times \log(x+1)$

Postfix:

Programming

$3 \times \log(x+1)$

$3 \times \log([x\,1+])$

operator: log; operand: $[x\,1+]$

$3 \times [x\,1+\log]$

op1: 3; op2: $[x\,1+\log]$

Postfix: $3\ x\ 1\ +\ \log\ x$

Tree

1) Binary tree
2) Complete binary tree
3) k-ary tree : Every internal node ⇒ k child

$$L + I = K \times I + 1$$

$$L = (K-1)I + 1$$

4) Full binary tree
5) $n_{max} = 2^{h+1} - 1$

   $n_{min} = h+1$

6) $n_{max} \Rightarrow 2^{h+1} - 1$ CBT, BT, BST, AVL tree

   $n_{min} \Rightarrow 2^h$ CBT

   $n_{min} \Rightarrow$ AVL

$$n(h) = 1 + n(h-1) + n(h-2)$$

7) BST : Inorder : Asc. order of keys.

8.) # unlabelled binary tree with n nodes $= \frac{2nC_n}{n+1}$

labelled binary tree with n nodes & n distinct keys $= \frac{2nC_n}{n+1} \times n!$

BST with n keys $= \frac{2nC_n}{n+1}$

Given a structure with n nodes & given n distinct keys

BT

BST $\Rightarrow$ 1

q. # Binary tree with a given preorder of length $n = \frac{2nC_n}{n+1}$

" " " Inorder " $= \frac{2nC_n}{n+1}$

" " " Postorder " $= \frac{2nC_n}{n+1}$

Shortcut / long In + Pre ⇒ tree
In + Post ⇒ tree

BST → skewed

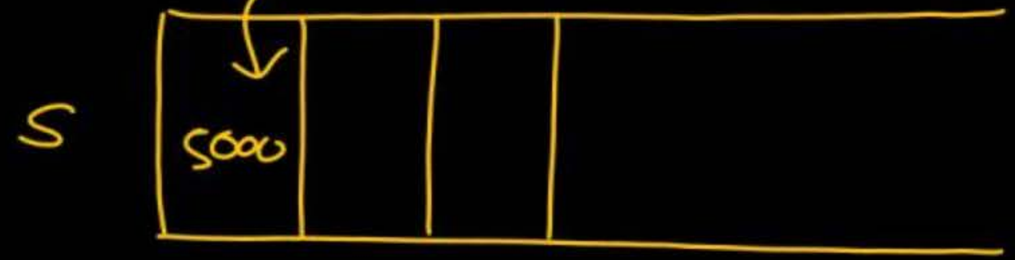
AVL tree

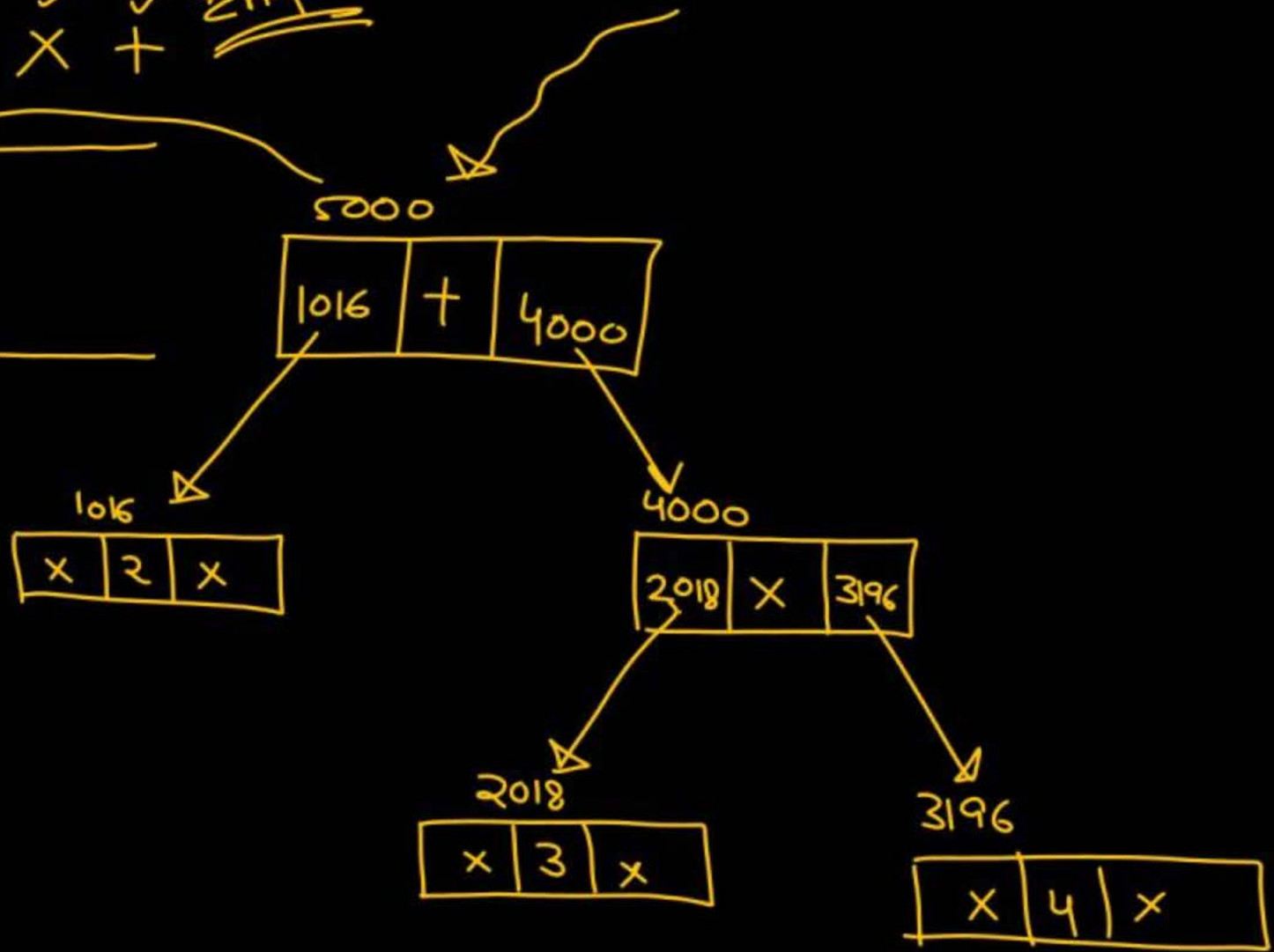
① BST Property

② AVL tree $-1, 0, +1$

infix : 2 + 3 x 4

Postfix: 2 3 4 x + End





inorder: 2 + 3 x 4

Pre : + 2 x 3 4

Post : 2 3 4 x +

THANK YOU GW
SOLDIERS !