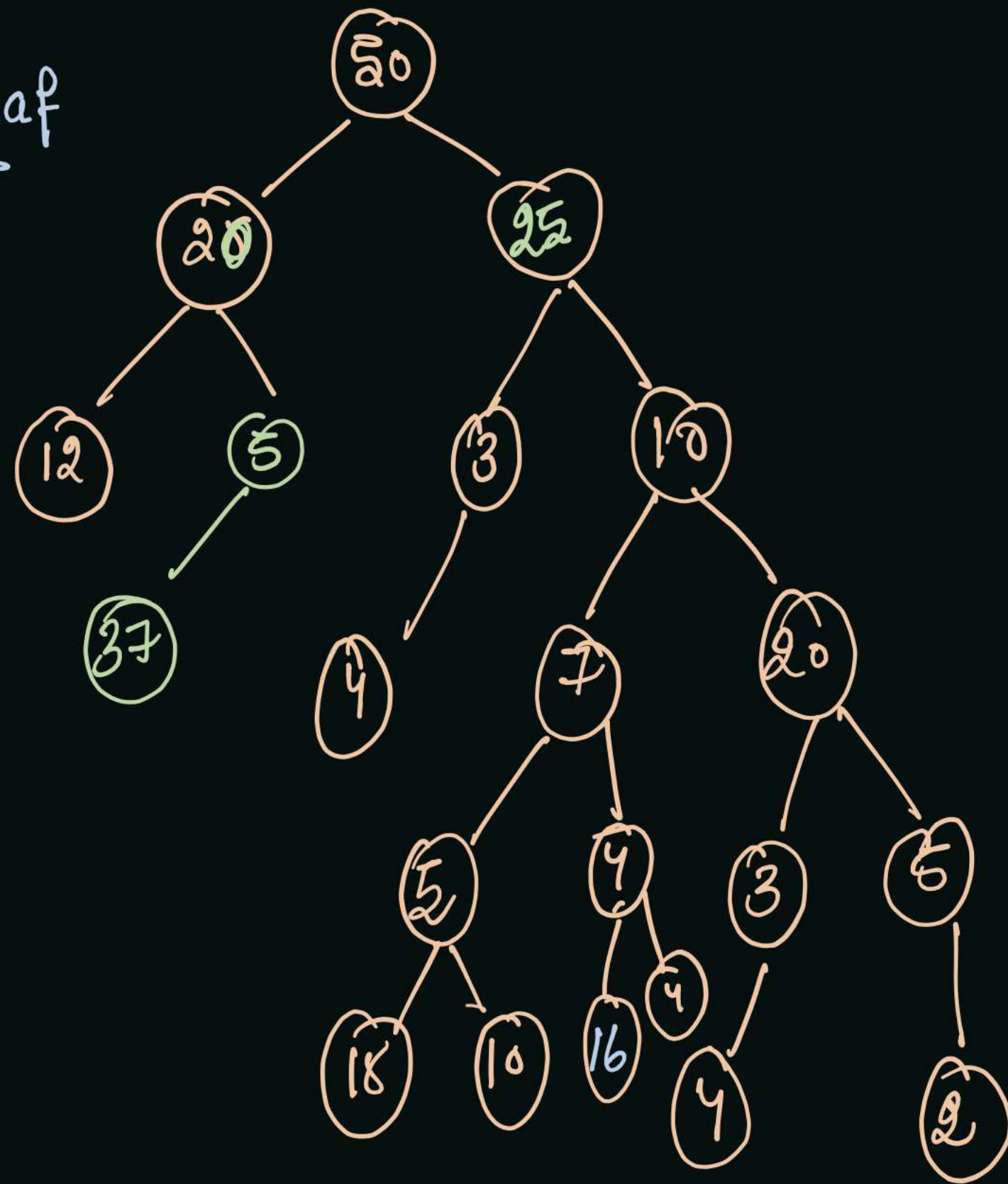


Path Sum in Binary Tree 2

Saturday, 28 August 2021 10:12 AM

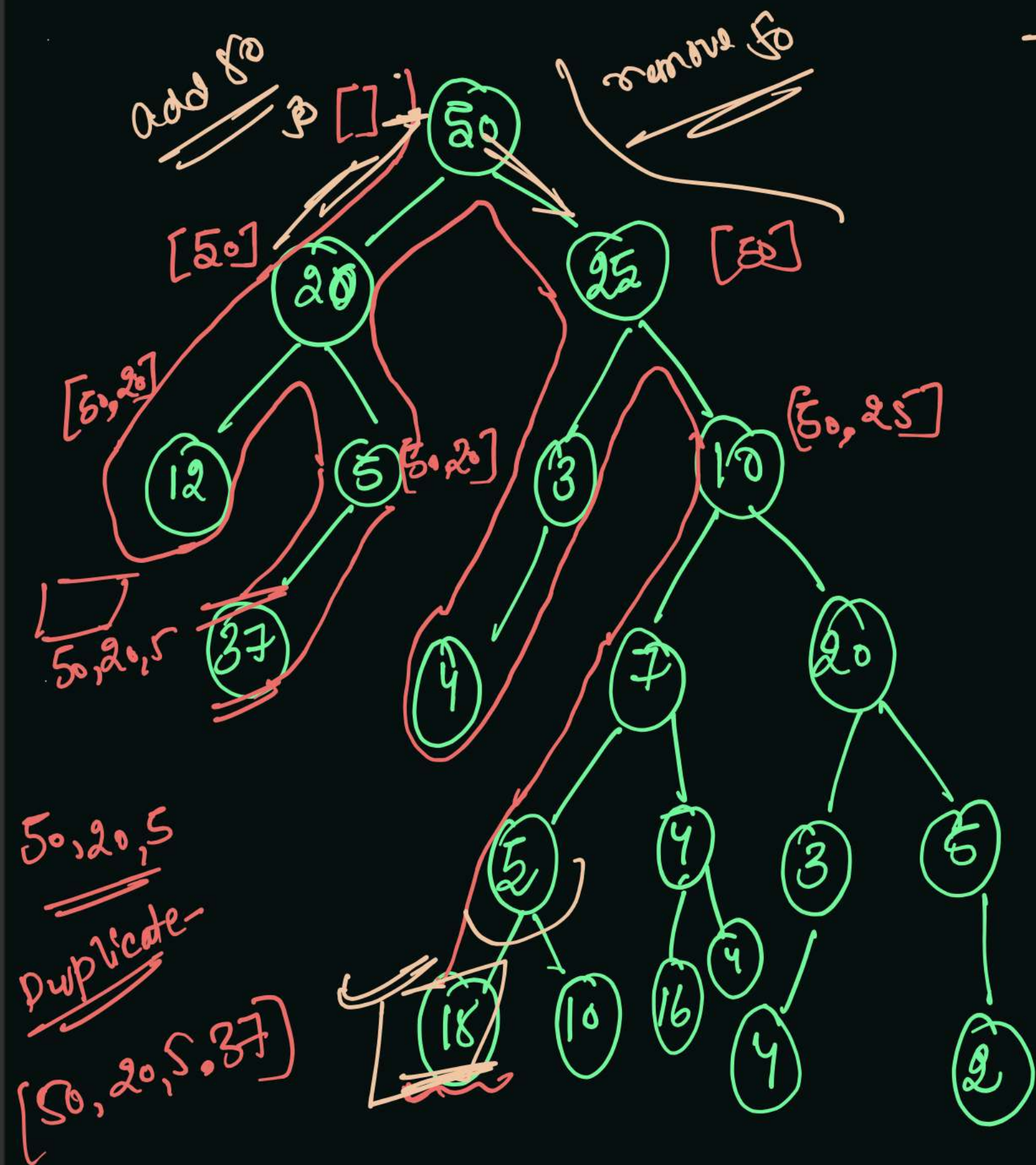
root to leaf
→

have sum
is equal
to target
sum



$$\text{target sum} = 50 + 25 + 37 = \underline{\underline{112}}$$

$$\begin{aligned} & [50, 20, 5, 37] \rightarrow 112 \\ & [50, 25, 10, 7, 5, 12] \rightarrow 112 \\ & [50, 25, 10, 7, 4, 16] \rightarrow 112 \\ & [50, 25, 10, 20, 3, 4] \rightarrow 112 \\ & [50, 25, 10, 20, 5, 2] \rightarrow \underline{\underline{112}} \end{aligned}$$



→ Res → [50, 20, 5, 37], [50, 25, 10, 7, 5, 18] ...
 SubRes → Path so far
 sum so far

```

if (root-left == null || root-right == null) {
  if (ssf + root-val == target sum) {
    // create a duplicate subarray
    and add val in it
    // add duplicate in final res
    resn
  }
}

```

50, 20, 5
 Duplicate →
 [50, 20, 5, 37]

50, 25, 10, 7, 5

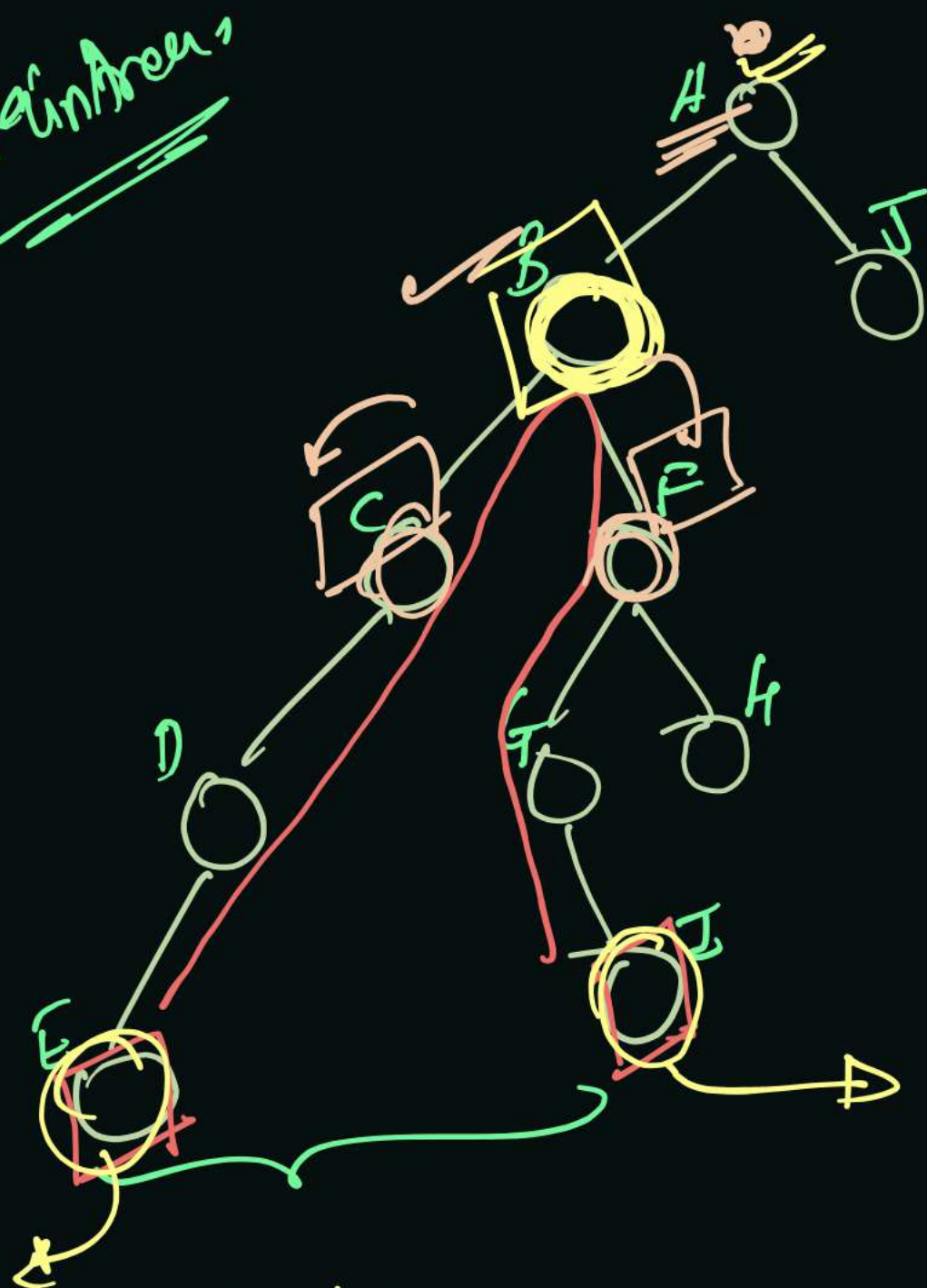
Duplicate → [50, 25, 10, 7, 5, 18]

Diameter of Binary Tree

Saturday, 28 August 2021

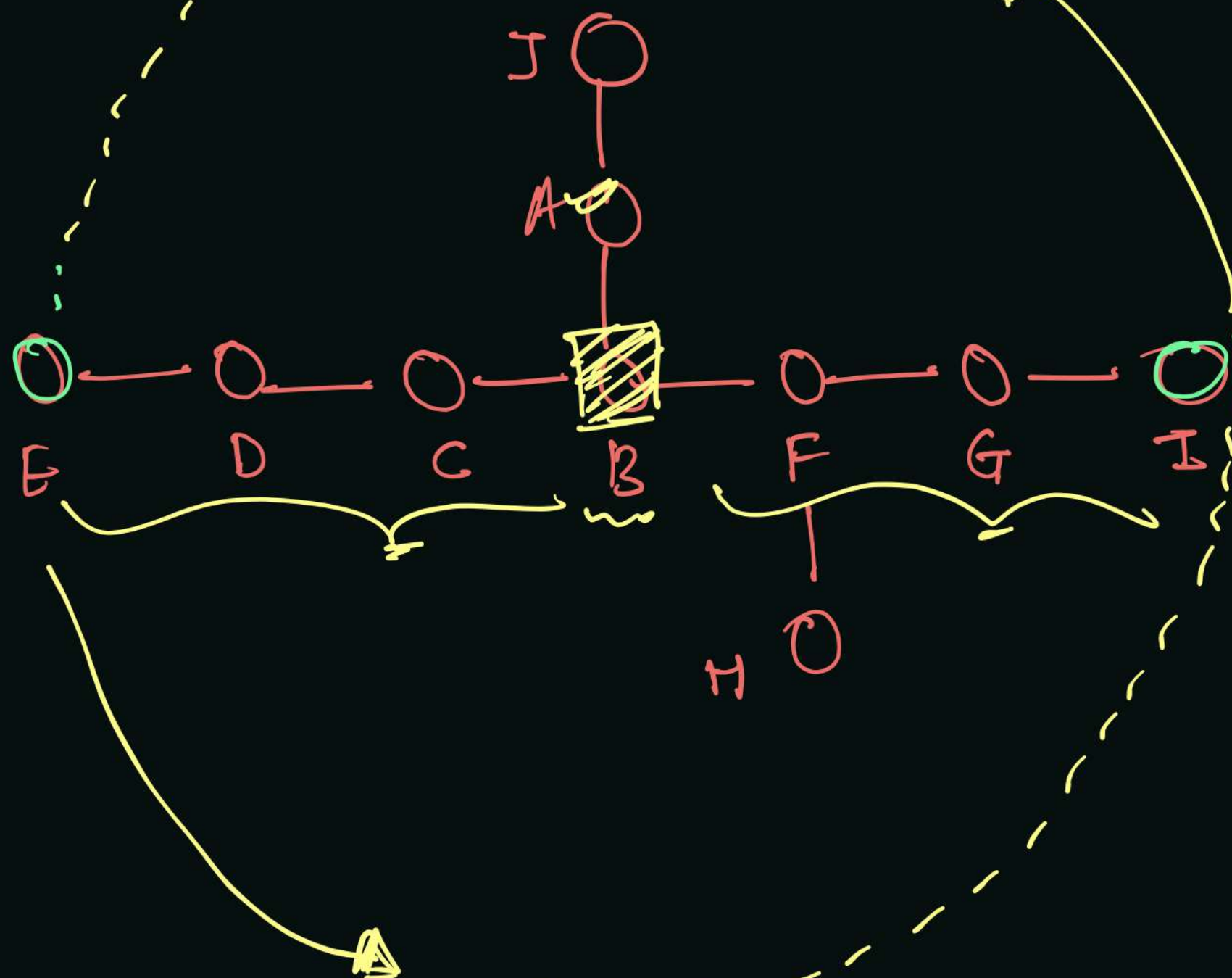
10:12 AM

main tree



stretch

Max distance between leaf \rightarrow E and I



Diameter = Diameter of tree.

diameter \Rightarrow Max distance between two leaf

write I \rightarrow calling with height
II \rightarrow static diameter variable and return height.
III \rightarrow Diameter

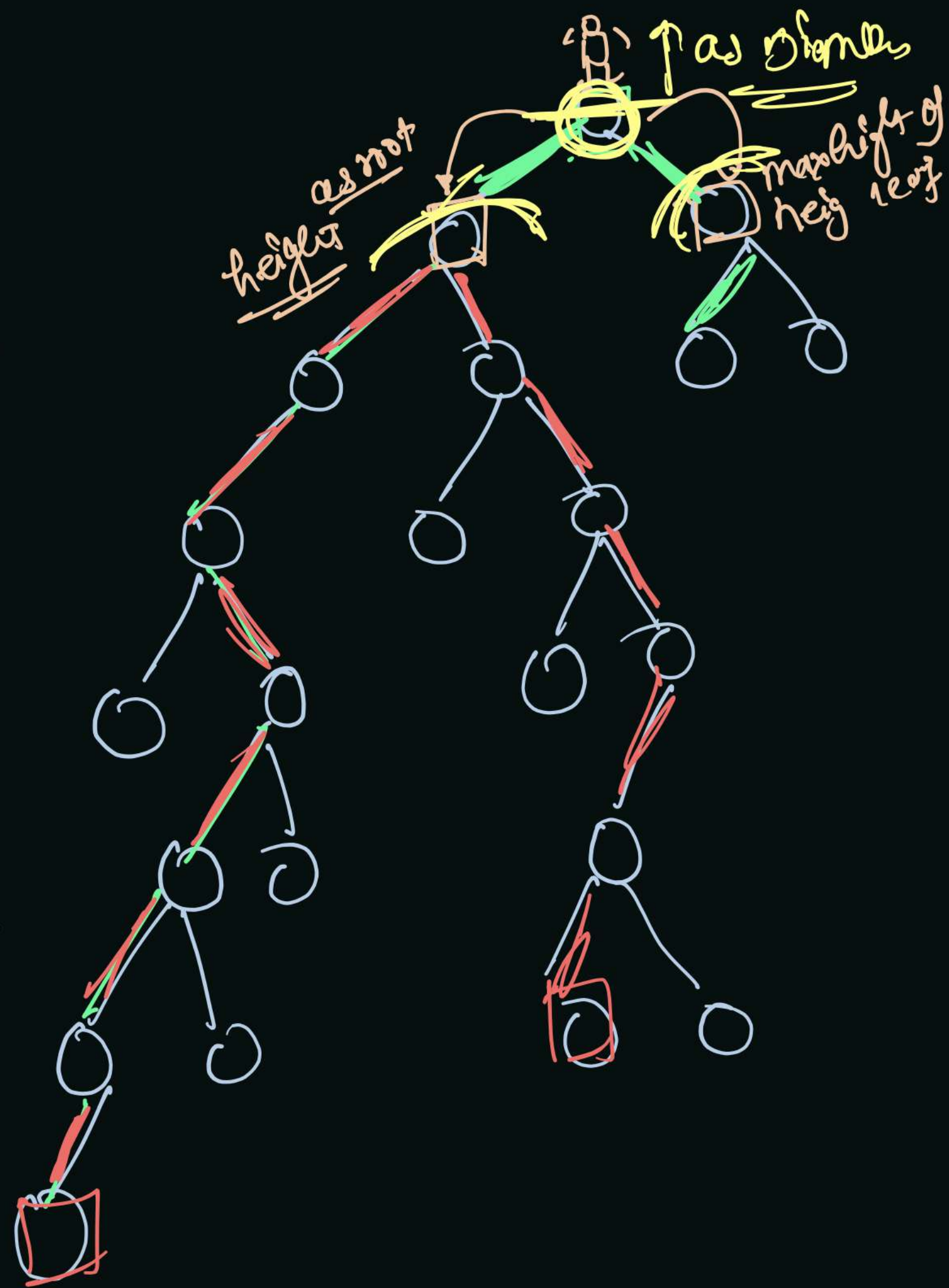
Area = πr^2 wrapper class

$$= \pi \left(\frac{d}{2} \right)^2$$

$$= \frac{\pi d^2}{4}$$

$$= 3.14 \times \frac{d^2}{4}$$

$$= \frac{3.14}{4} \times \frac{d^2}{1}$$



Two max distance Leaf
If

→ Diameter is passing through Root

$$\text{dia} = \text{left height} + \text{right height} + 2$$

$$= 6 + 1 + 2 = \boxed{9}$$

Left subtree have also a diameter

Similarly, Right subtree have also a diameter

computation →

Left subtree diameter

vs.

Right subtree diameter

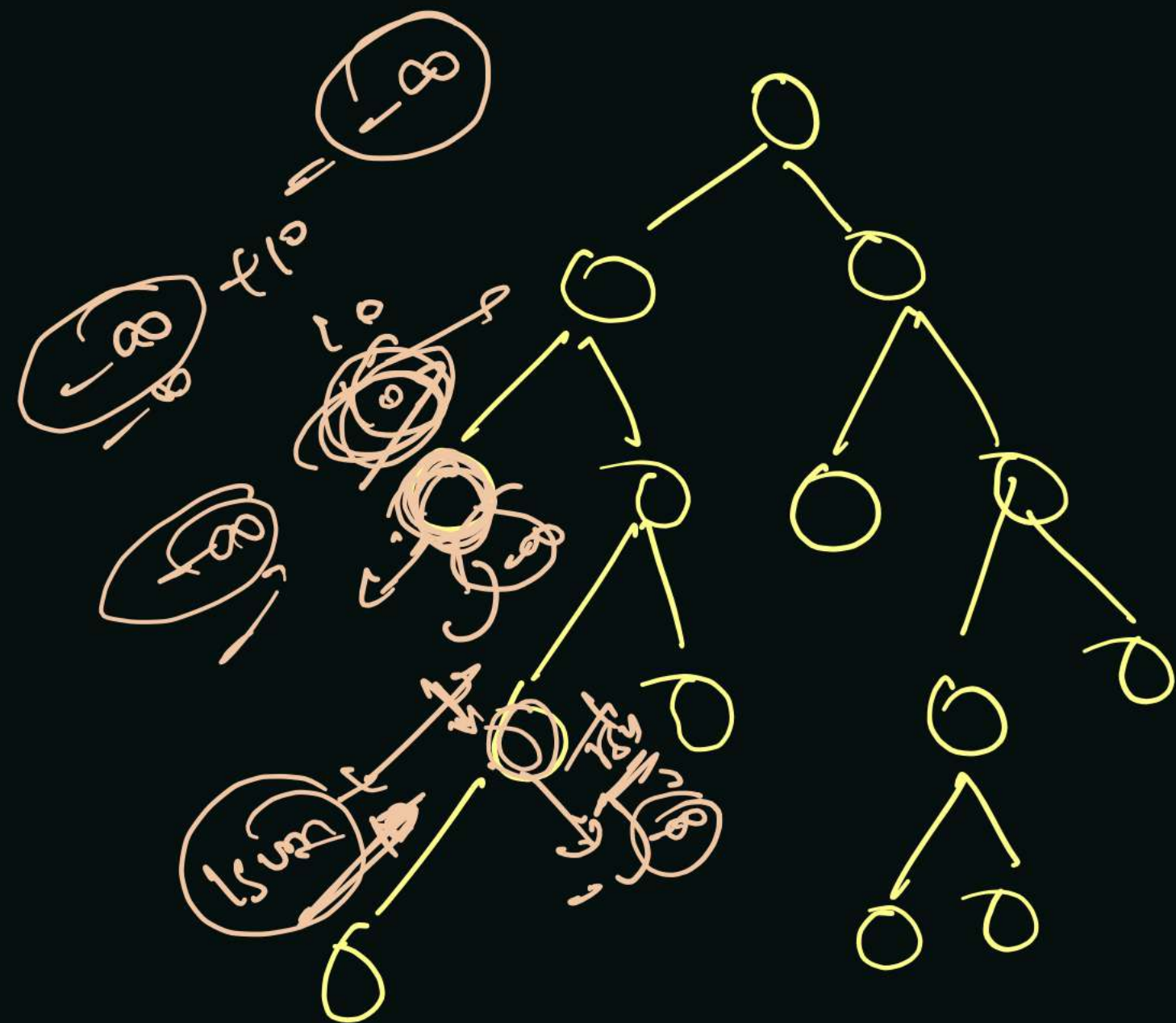
vs

Diameter passing through Root

max

Maximum Path Sum in Between Two Leaf

Saturday, 28 August 2021 10:12 AM



Two leaf (path sum) \equiv Diameter of Tree

Dependency on Max sum
Between Root to
leaf

Dependency on height

(I) If max path is passing through Root \rightarrow
left \rightarrow Root to leaf max path
right \rightarrow Root to leaf max path
path sum = $\underbrace{\text{left} + \text{right} + \text{root-val}}$
Root is part of path.

three Methods

\hookrightarrow (1) Node to leaf path

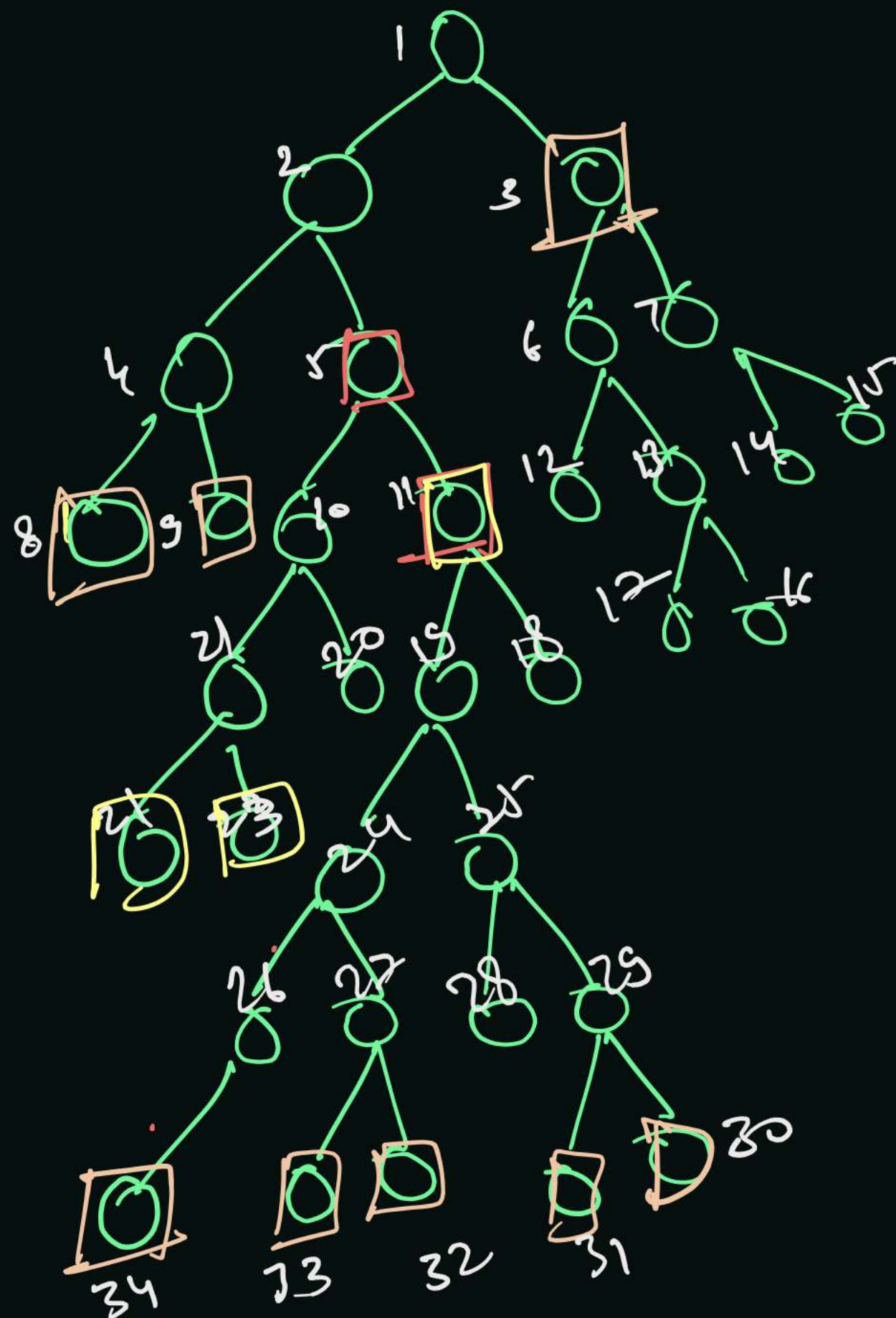
~~(2) static~~ \rightarrow Return type \rightarrow Dependency

(14) wrapper class \hookrightarrow static variable \hookrightarrow Result

(II) [left subtree \rightarrow max path b/w two leaf

(III) [right subtree \rightarrow max path b/w two leaf

K far,



$k=4$

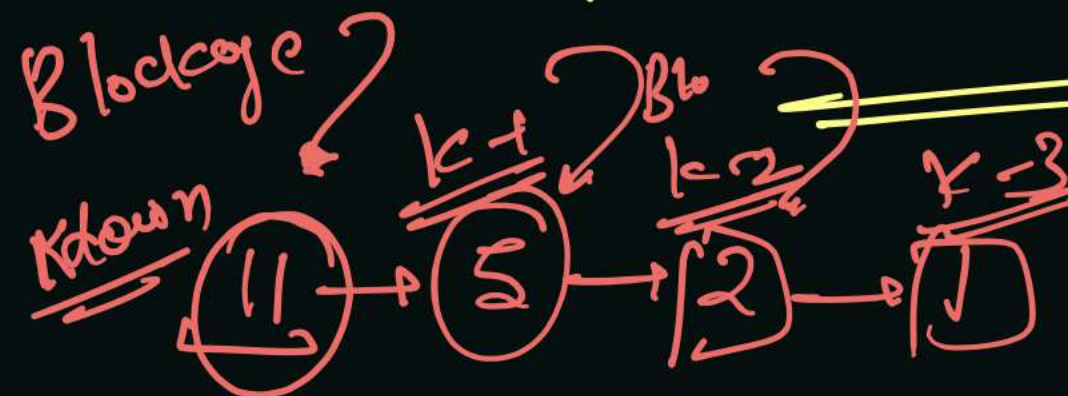
node = ()

$$[34, 33, [32, [81, 30], 21, [23, 8, 9, 3]]]$$

Requirement \rightarrow kdown

Node to Root path $\langle \text{Node} \rangle$ type

Blockage



37 33 32 31 21 23 8 9 3



Leaf complete Tree

Adjacent node will catch fire ==

Path of Node \rightarrow Time

Bun \rightarrow

$$\begin{bmatrix} 0 & \rightarrow & \text{---} & & & \\ 1 & \rightarrow & \text{---} & \text{---} & \text{---} & \\ 2 & \text{---} & \text{---} & \text{---} & \text{---} & \end{bmatrix}$$