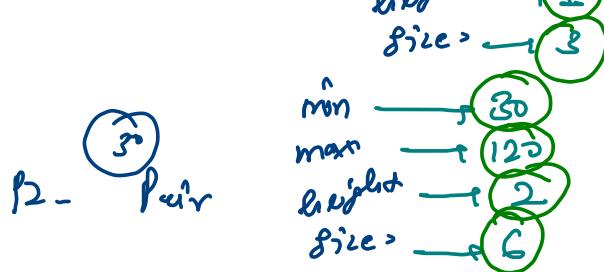
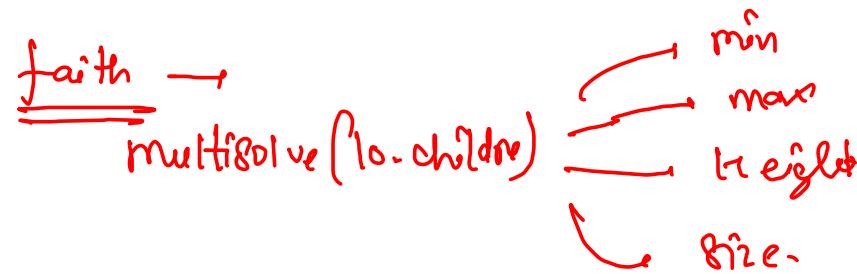
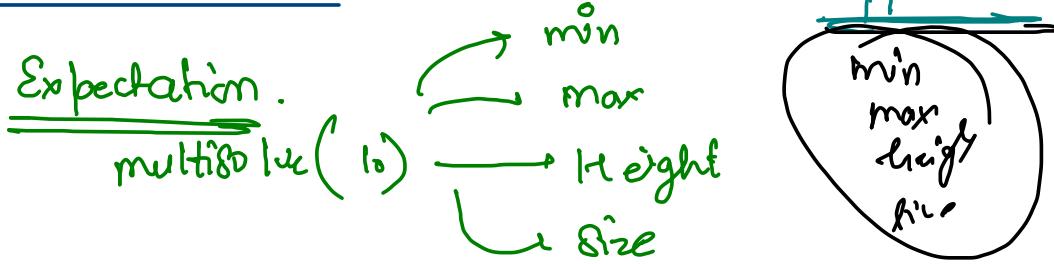
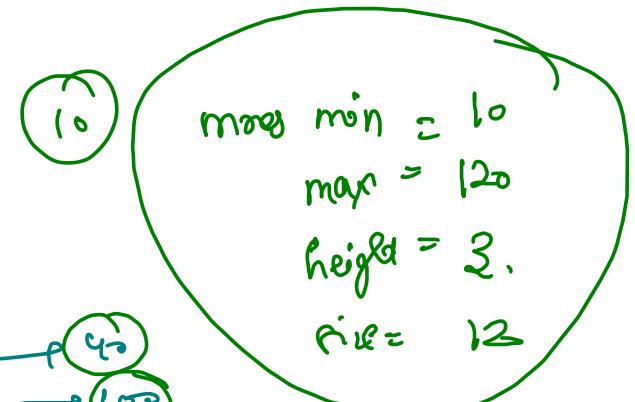
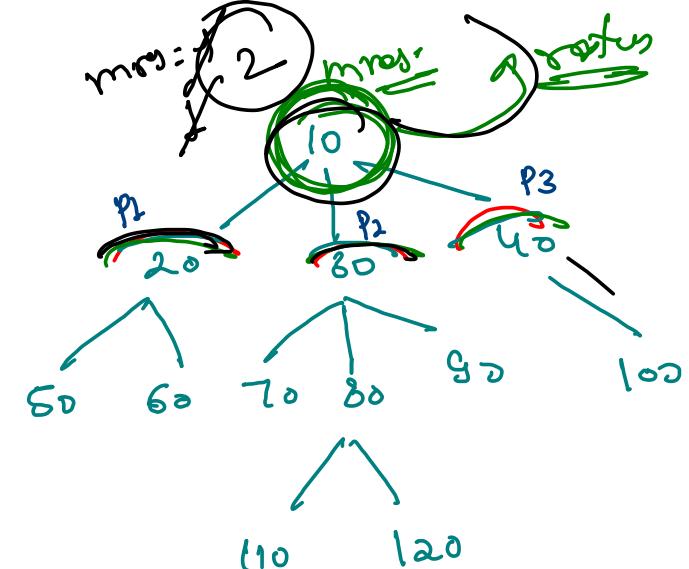
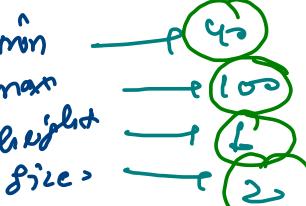


Multisolver Technique



P3 - $\begin{matrix} 40 \\ \text{Pair} \end{matrix}$



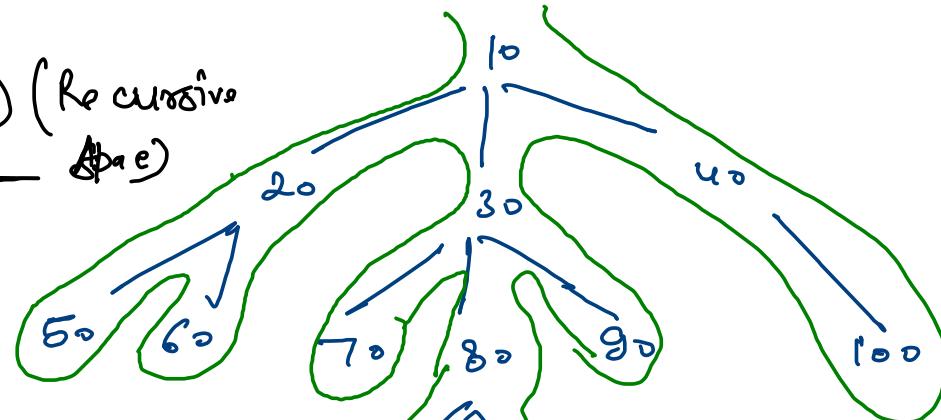
Predecessor and Successor

data = 80

Predecessor → 70

Successor. → 110

Time → $O(n)$
Space → $O(h)$ (Recursive stack)



data = 40

PreOrder Predecessor → ?

Successor. → ?



PreOrder → just previous element of data in preOrder ↗
 Predecessor and after data encountered. element is successor

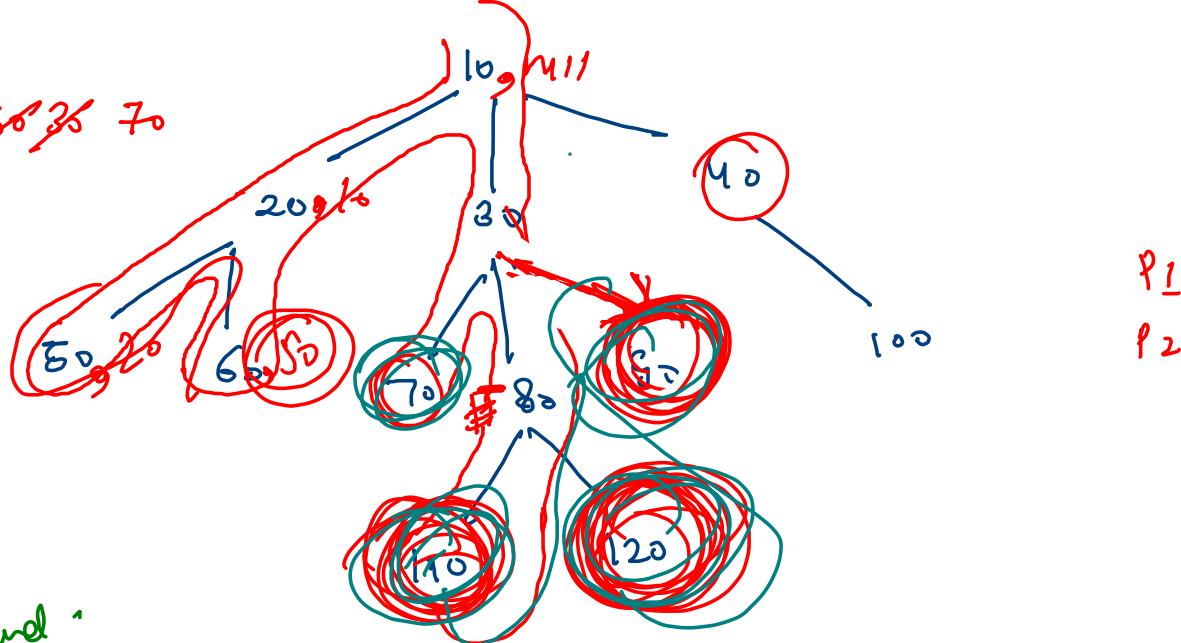
~~pred = null~~ 16 20 56 60 36 70
~~succ = null~~ 110

State: σ_1

data = 80

State $\approx \sigma$

data Not found



State $\approx \sigma$

first node after data found.

State $\sigma \approx \sigma$

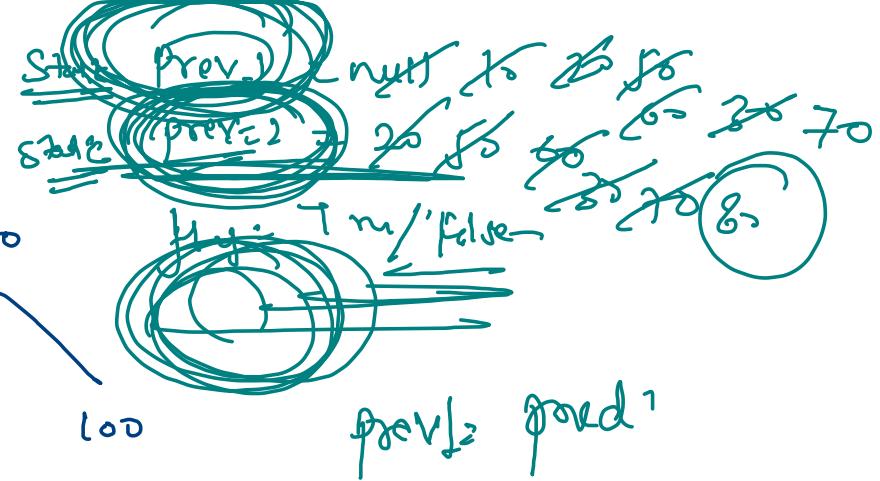
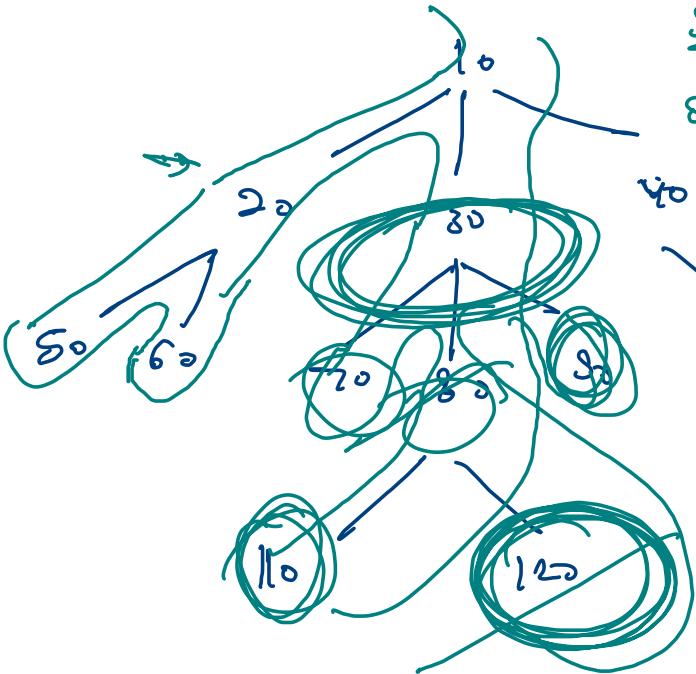
return

data = 120



Approach 2

datu = 8



Node = 82

Ceil and Floor →

node. data is considered as Number.

$$\text{factor} = \underline{\underline{70}}$$

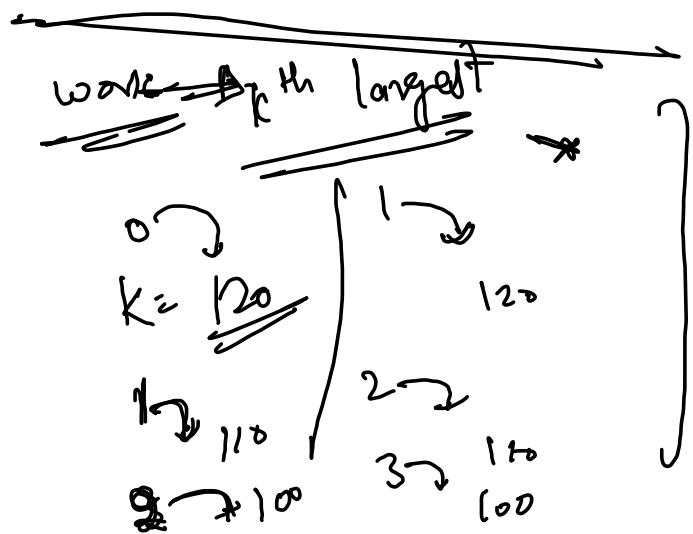
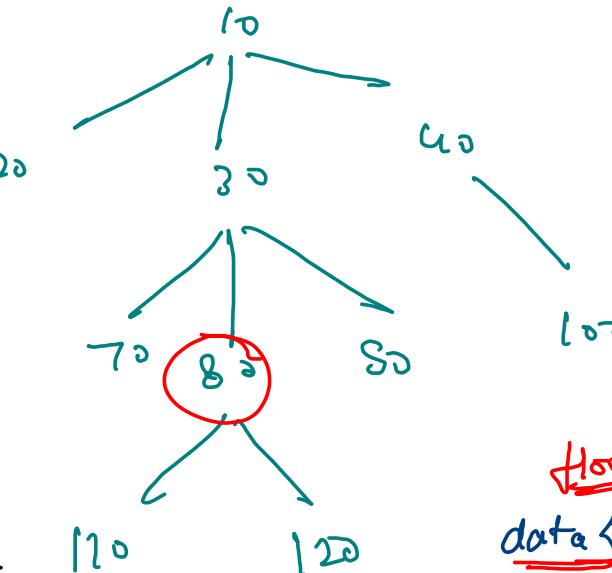
$$\text{ceil} \rightarrow ? \rightarrow 80$$

$$\text{Floor} \rightarrow ? \rightarrow 60$$

just greater than factor → ceil → qualified min

floor → qualified max

just smaller than factor



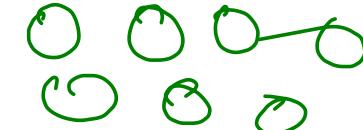
Floor → data < factor → ceil → data > factor.

valid data set for floor of factor



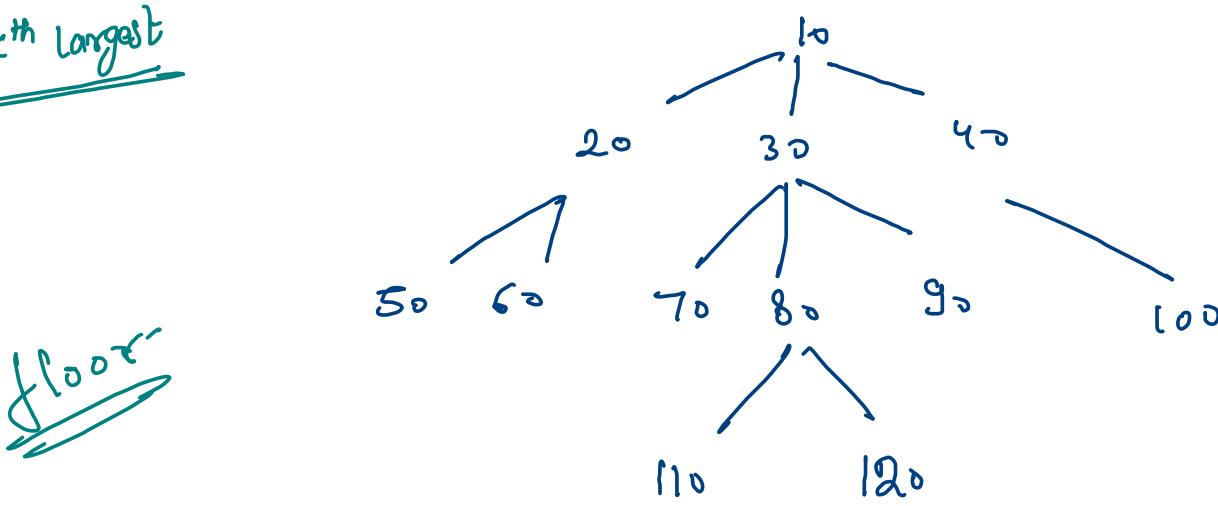
winner → max

valid data set for ceil of factor



winner → min

k^{th} largest

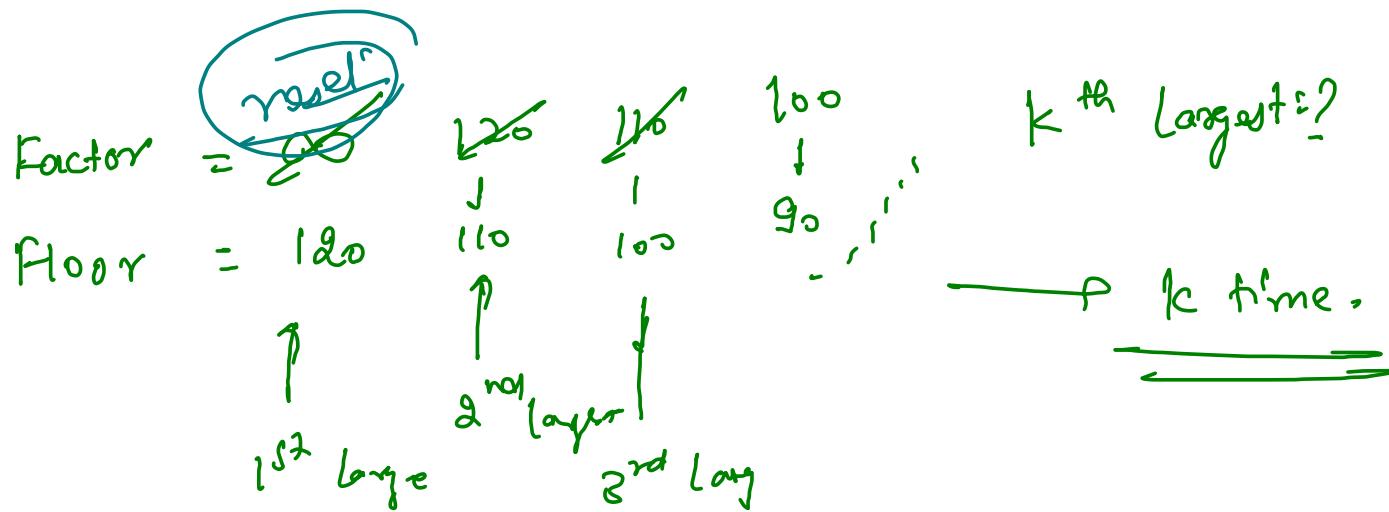


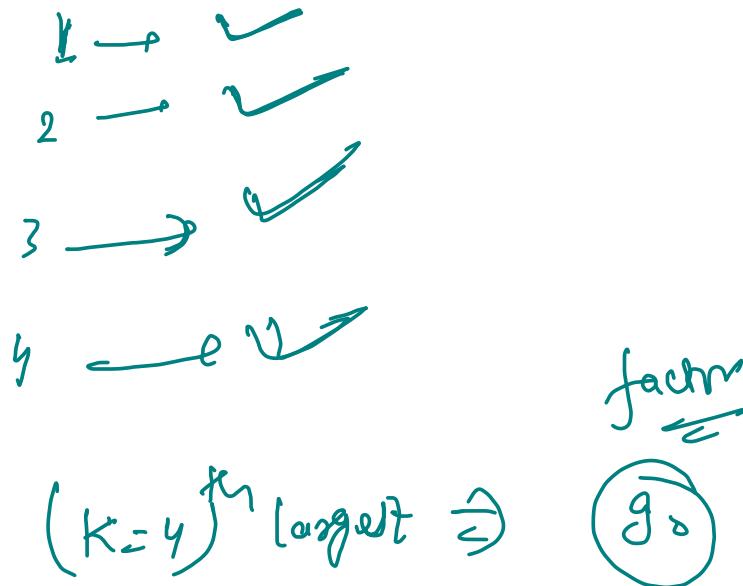
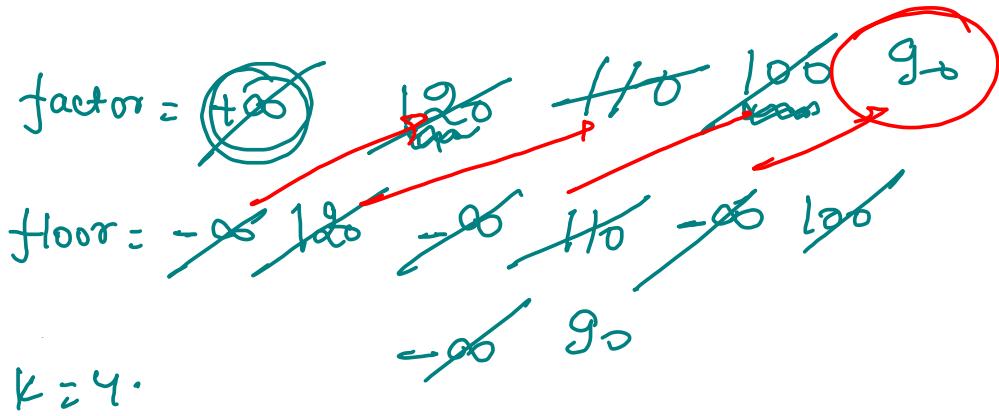
1st largest = 120

2nd largest = 110

3rd largest = 100

⋮
⋮
⋮
10th largest =

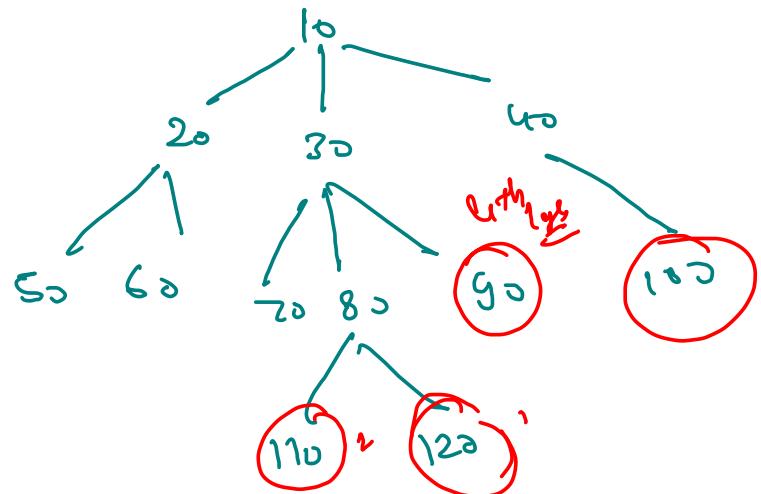




```
public static int kthLargest(Node node, int k){
    int factor = Integer.MAX_VALUE;

    for(int i = 0; i < k; i++) {
        floor = Integer.MIN_VALUE;
        ceilAndFloor(node, factor);
        factor = floor;
    }

    return factor;
}
```



Node with maximum Subtree Sum.

① Subtree sum

Node having

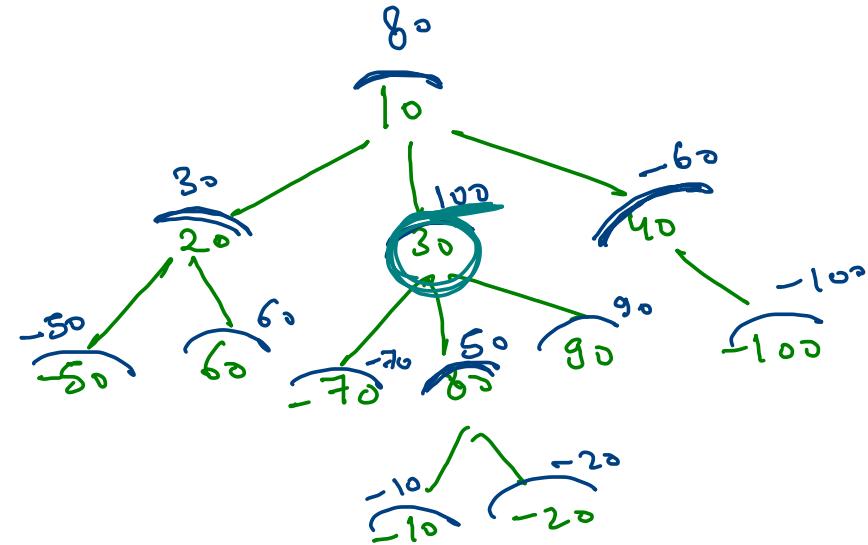
maximum
subtree
sum.

$30 @ 100$

Node @ sum

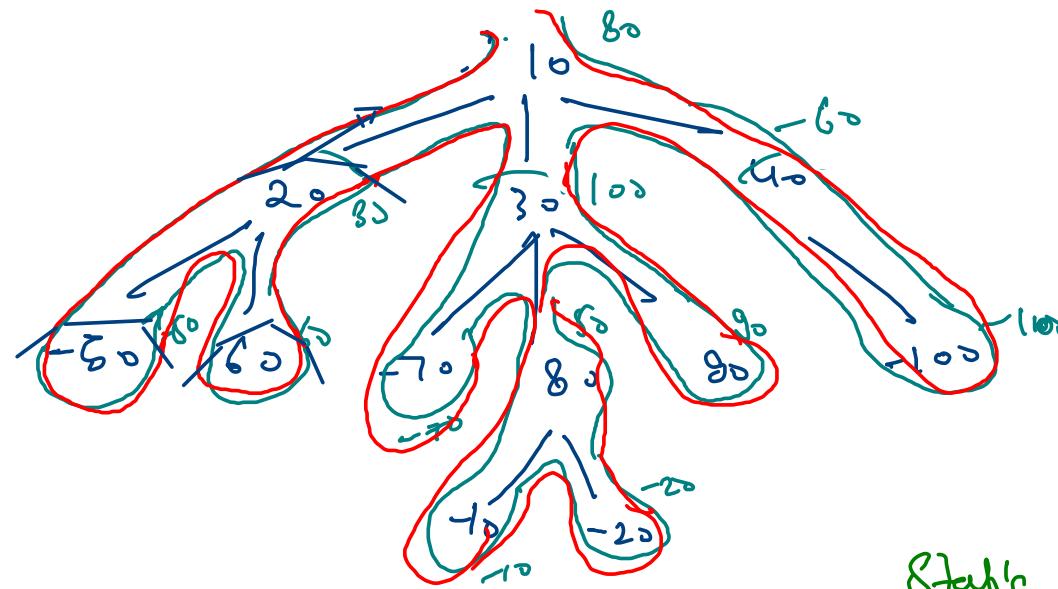
Hints: static variable g

Node and sum.



Sum of Node, data

Sum of Tree



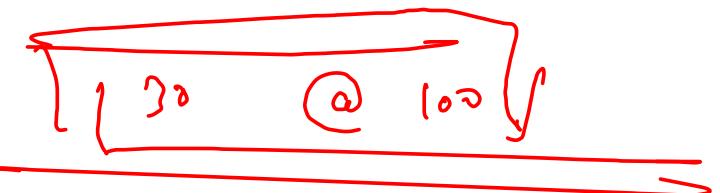
Subtree
Post Area → Tree
Sum

Expectation →
 $\sum_{i=1}^{16}(i)$ →
Sum of all
node
including 16

faith →
 $\sum(16 \cdot \text{child})$

Static Variable -

node → 100 60 90 30
sum = → 100 60 90 100



Diameter of Tree : Distance between two farthest Node in tree

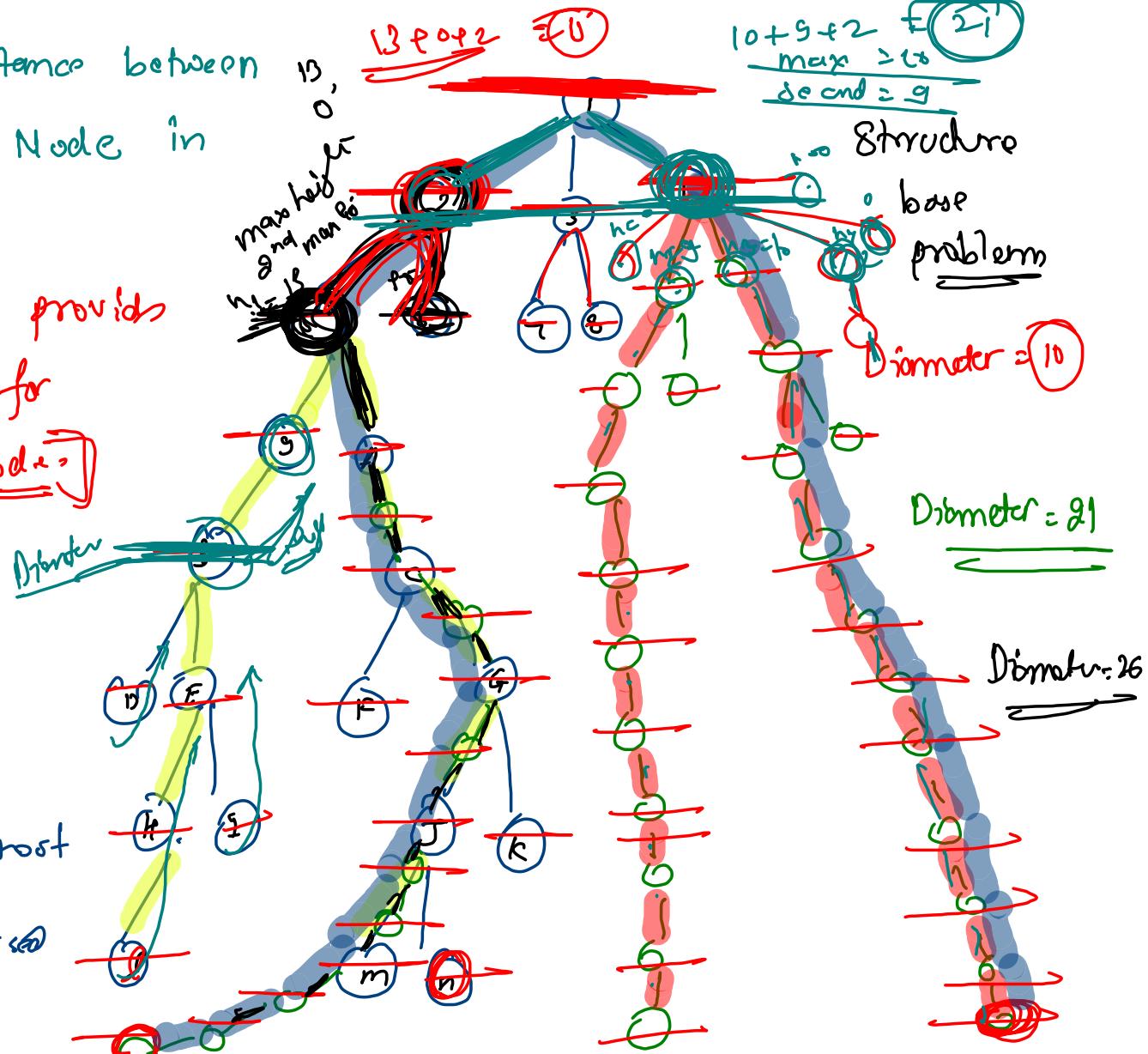
Two points
which have
max dist.
b/w them

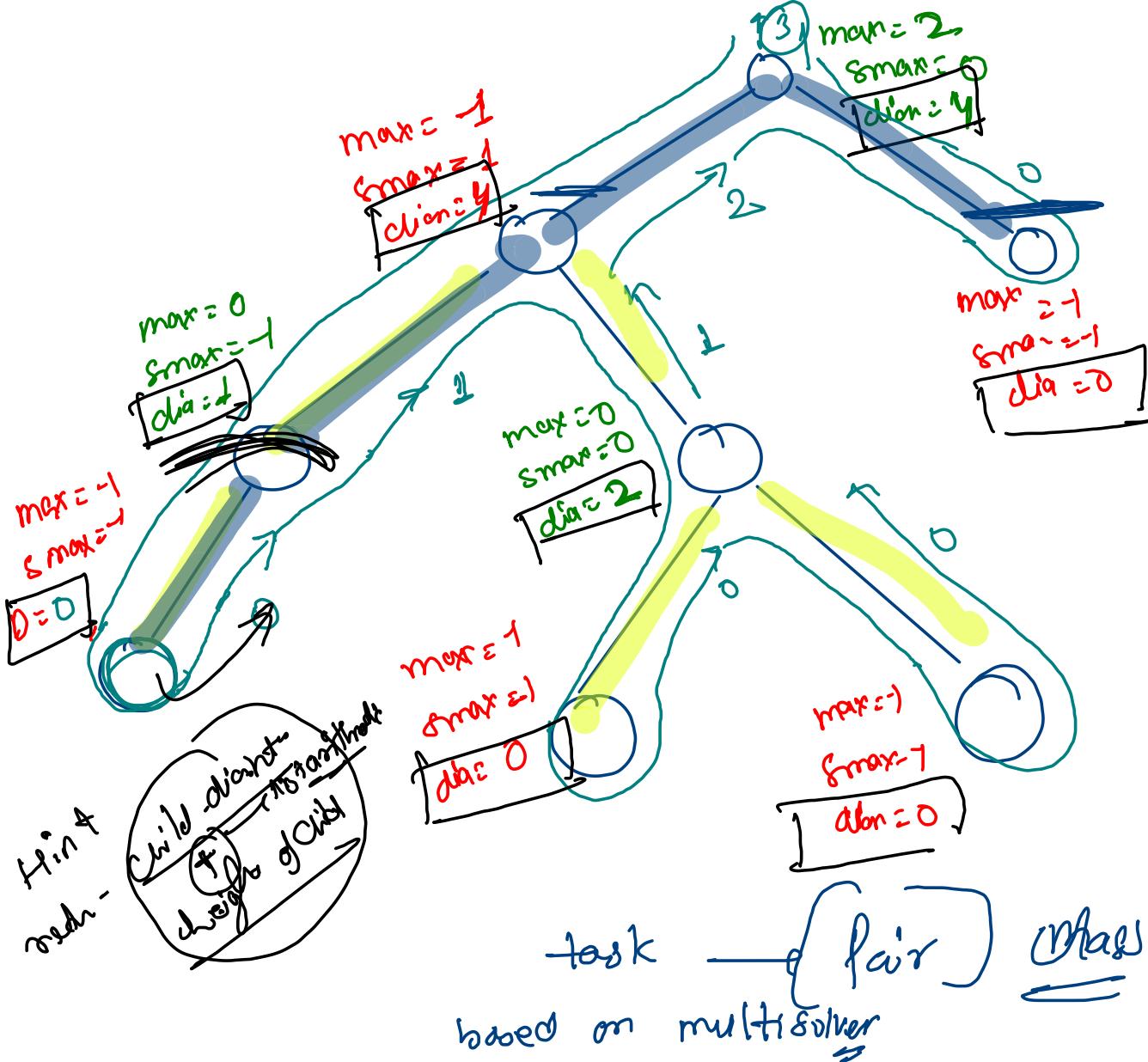
solution → provides opportunity for every mod- \rightarrow

~~(1)~~ Diameter can be on
left side of
root

~~✓~~ Diameter can be seen
right side of nose

 Diameter can be passed
for root.





height → Return

prepare →
 max height
 second max height

\max^{max}
 diameter = 4

Complexity: $O(n)$

→ without static
variable