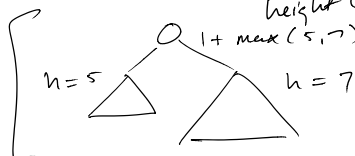


height(t) = maximum depth of any node

$$height(t) = \begin{cases} 0 & t = \phi \\ 1 + \max(height(left), height(right)) & \text{otherwise} \end{cases}$$



class tnode

```

{ public:
    int data;
    tnode(int d,
           tnode* L = NULL,
           tnode* R = NULL)
    {
        data = d;
        left = L;
        right = R;
    }
    tnode* left;
    tnode* right;
};

```

int main()

```

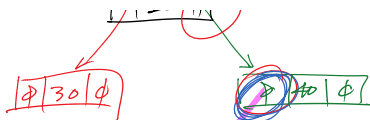
{
    tnode* t = NULL;
    t = new tnode(10,
                  tnode(20,
                        tnode(30,
                              tnode(40));
    t
}

```









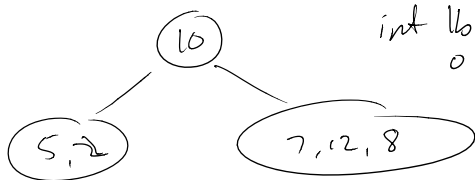
$t = \text{new} (10, \text{tnode}(20, \downarrow, \downarrow))$



$((t \rightarrow \text{left}) \rightarrow \text{right}) \rightarrow \text{left} =$   
 $\text{new tnode}(30);$

$d = \begin{array}{|c|c|c|c|c|c|} \hline 0 & 1 & 2 & 3 & 4 & 5 \\ \hline 5 & 2 & 10 & 7 & 12 & 8 \\ \hline \end{array} \quad n=1$   
 $\frac{0+5}{2} = 2$

$\text{tnode}^* \text{makeTree}(\text{int } d[], \text{int } n)$   
 $\text{int } lb, \text{int } ub$   
 $0 \quad n-1$



$\text{tnode}^* \text{makeTree}(\text{int } d[], \text{int } lb, \text{int } ub)$

$\{$

$\text{if} (lb \leq ub)$

$\{$

$\text{int } m = (lb + ub) / 2;$

$\text{tnode}^* L = \text{makeTree}(d, lb, m-1);$

$\text{tnode}^* R = \text{makeTree}(d, m+1, ub);$

$\text{return new tnode}(d[m], L, R);$

$\{$

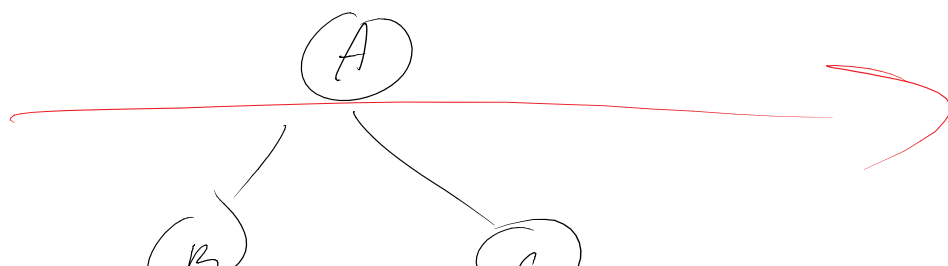
$\text{else}$

$\{$

$\text{return NULL};$

$\{$

$\}$

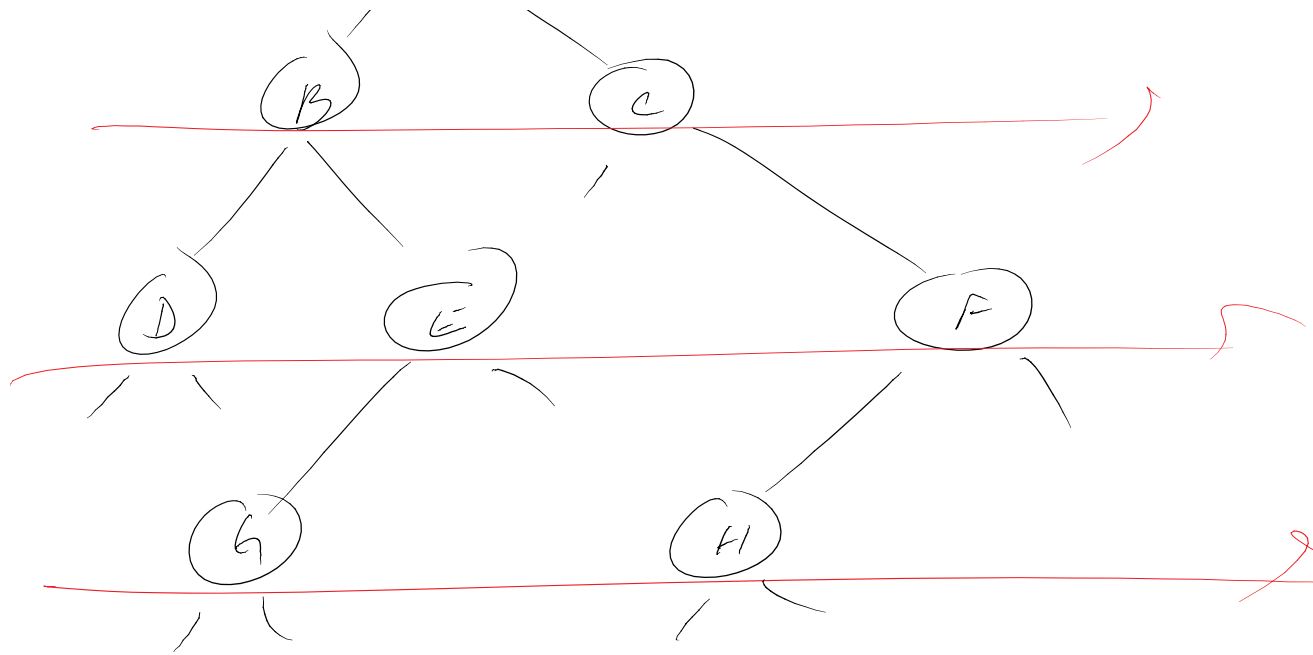












pre order: Root, left pr

A, B, D, E, G, C,

post order: Left /

D, G, E, B,

in order: Left

D, B, G, E, A

e, Right Pre

F, H

post, Right post, root

H, F, C, A

- in, root Right ir

, C, H, F

7

9



D, B, C, E, H

Level-by-level

A, B, C, D,

m

Write a

the height

int h

, C, H, I

ed :

E, F, G, H

---

Function to compute  
of a binary tree  
height (tnode \* t)

---

$\frac{1}{x}$

$\frac{1}{x}$

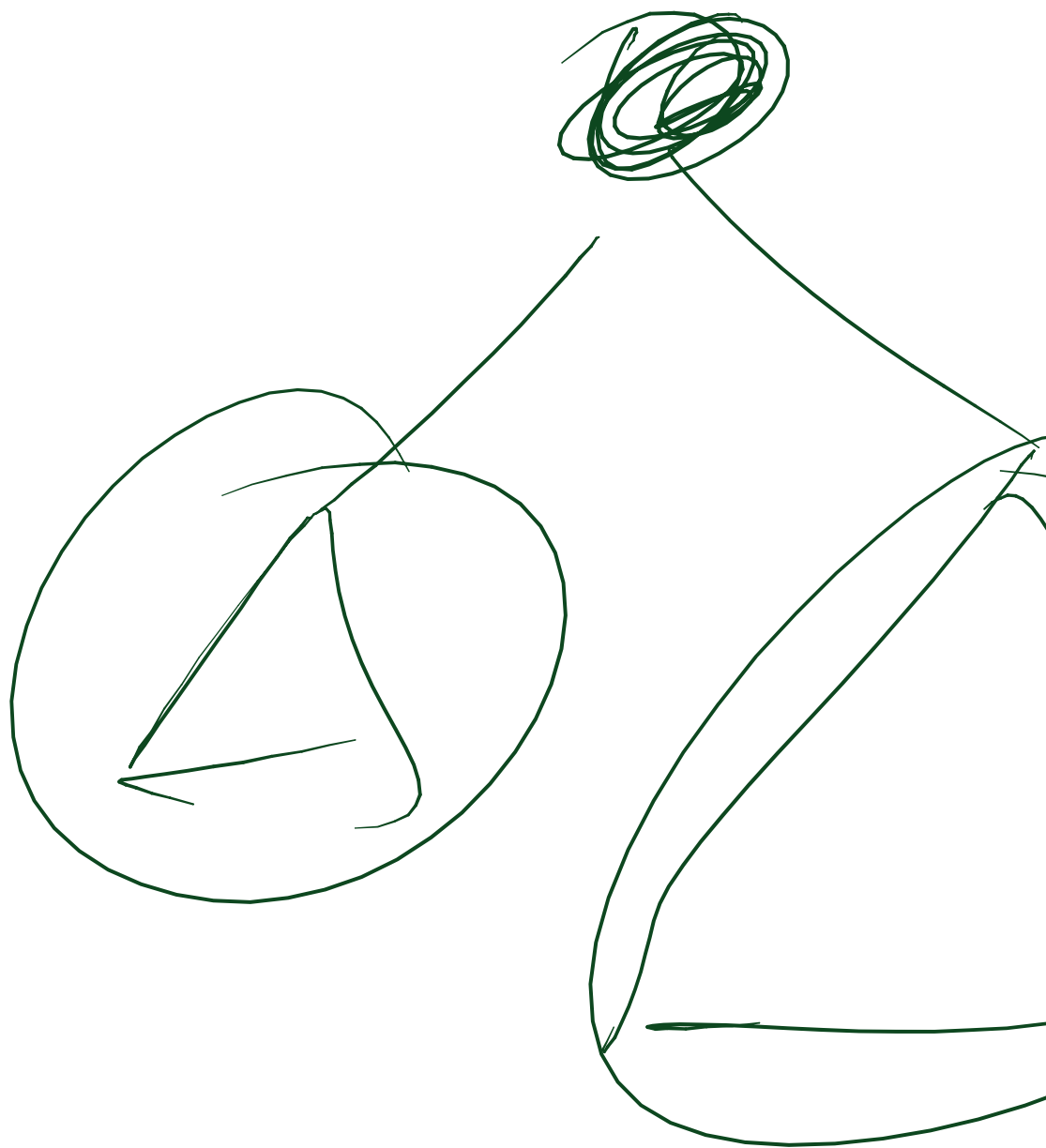


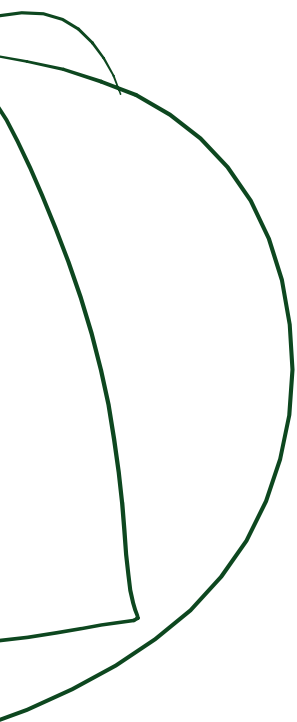


---

```
int height  
{  
    if (t  
        re  
    else  
        retu
```







3

height (  $t \rightarrow$  left

height (  $t \rightarrow$  n

),  
glt),

