1 2 0

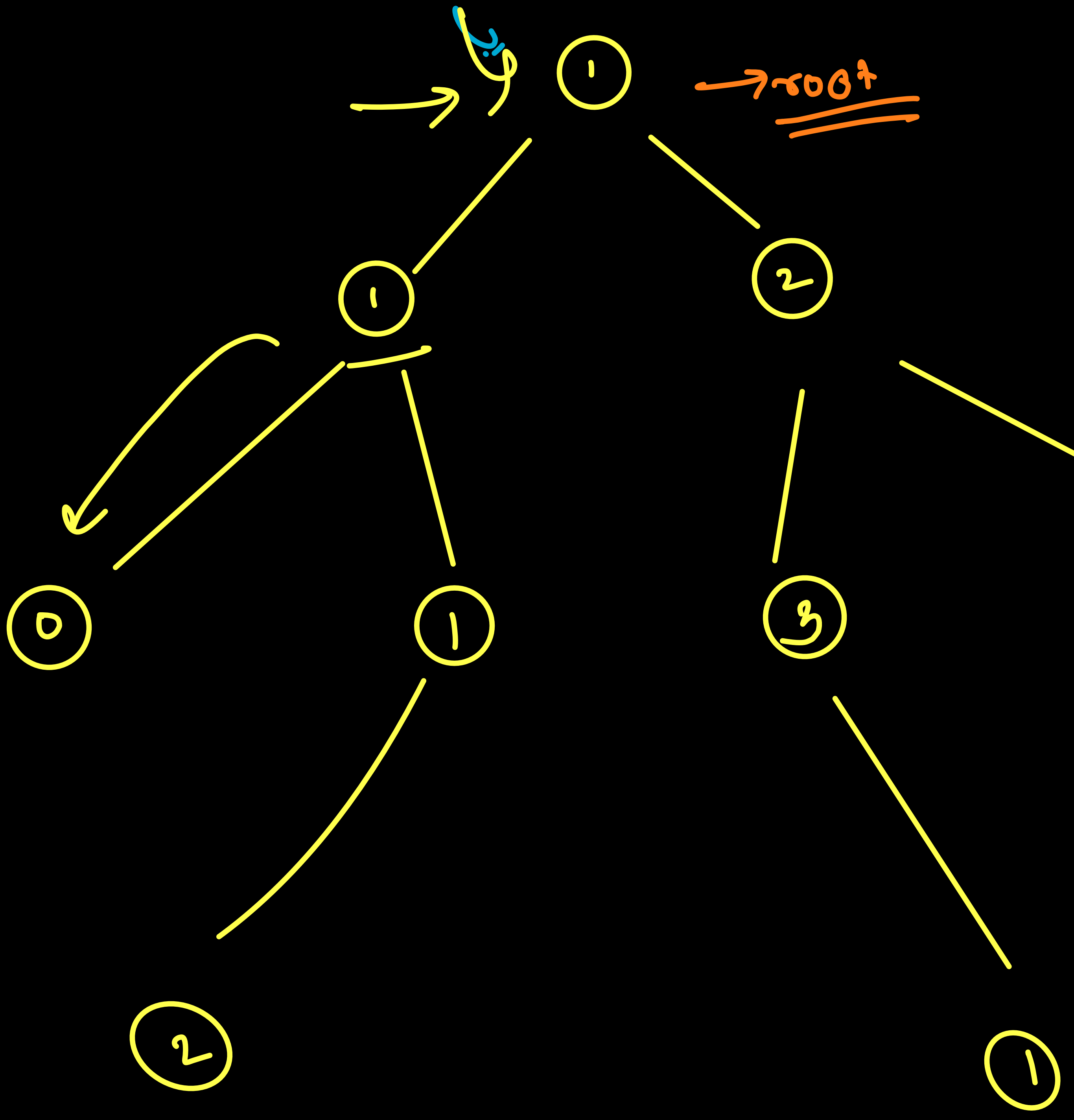
1 1 0

1 1 1 2

1 2 3

2 5 7 3

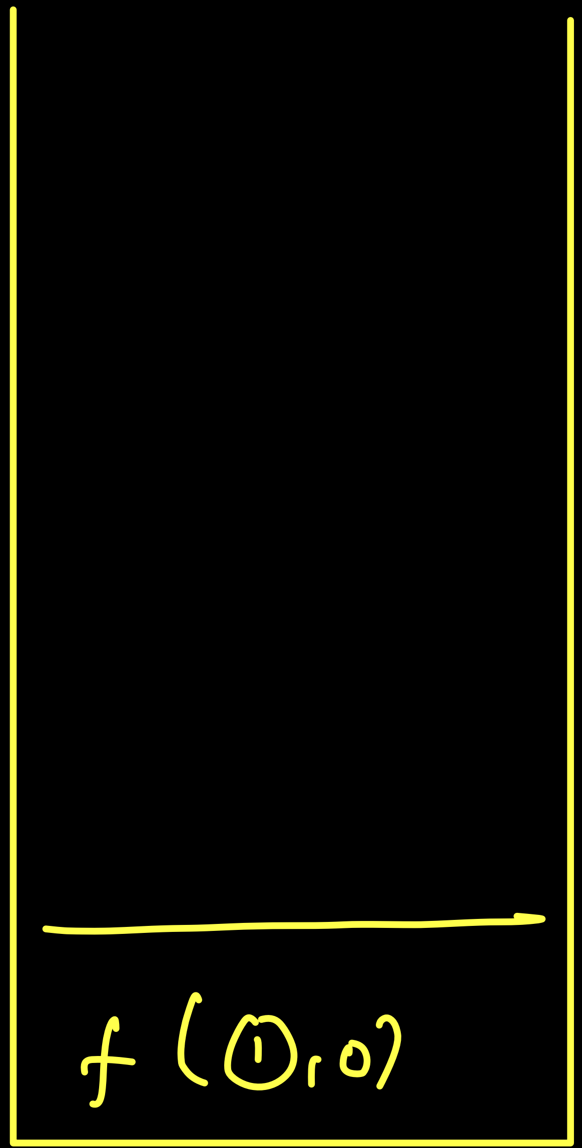
ans



1, 0

1, 1

0, 11



ans → 0 f 110 → 110 f 1112

JOIN THE DARKSIDE

f(root, $\overset{\nearrow 0}{num}$) ⟹

```
{
    number = 10 * num + root.val
    if(root.left){
        f(root.left, number)
    }
    if(root.right){
        f(root.right, number)
    }
}
```

↓

travel the tree in dfs fashion & store the root → leaf value in num

ans → 0

global

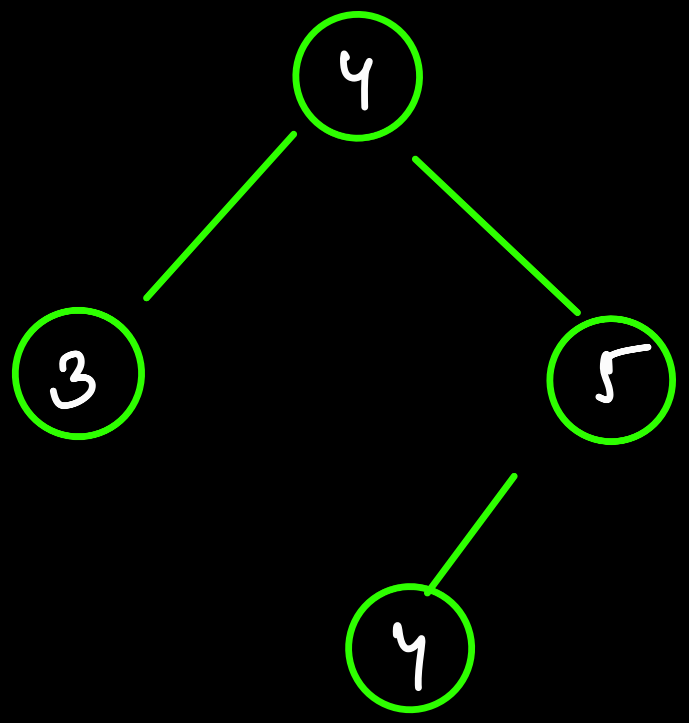
```
if ( root.left == null && root.right==null){
    ans += (10 * num + root.val)
    return
>
```

JOIN THE DARKSIDE

how to uniquely identify a subtree ??

2 subtree if they are same have the same order & value of nodes

preorder → string



→ 4, 3, #, #, 5, 4, #, #, #

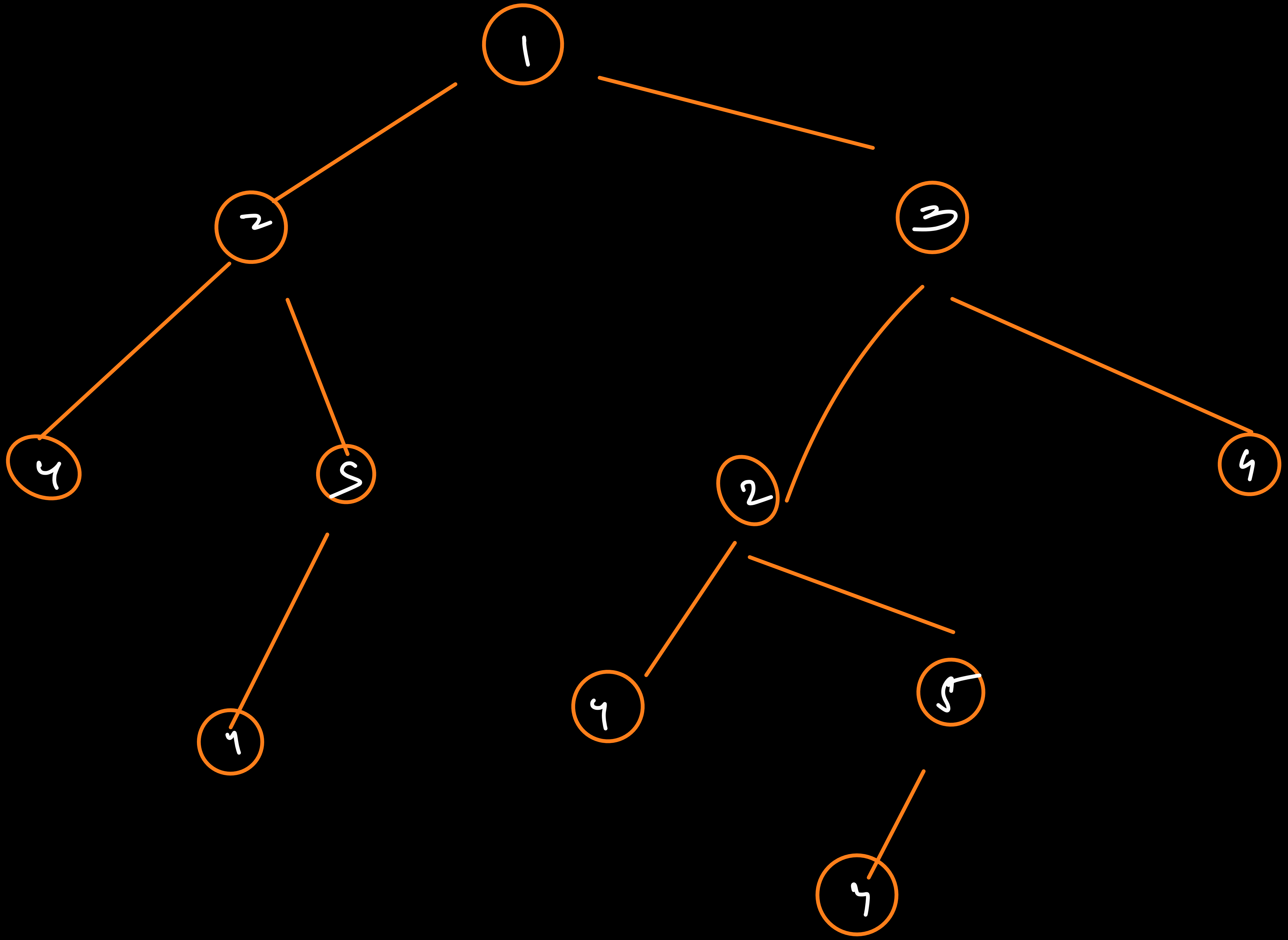
{

str : freq

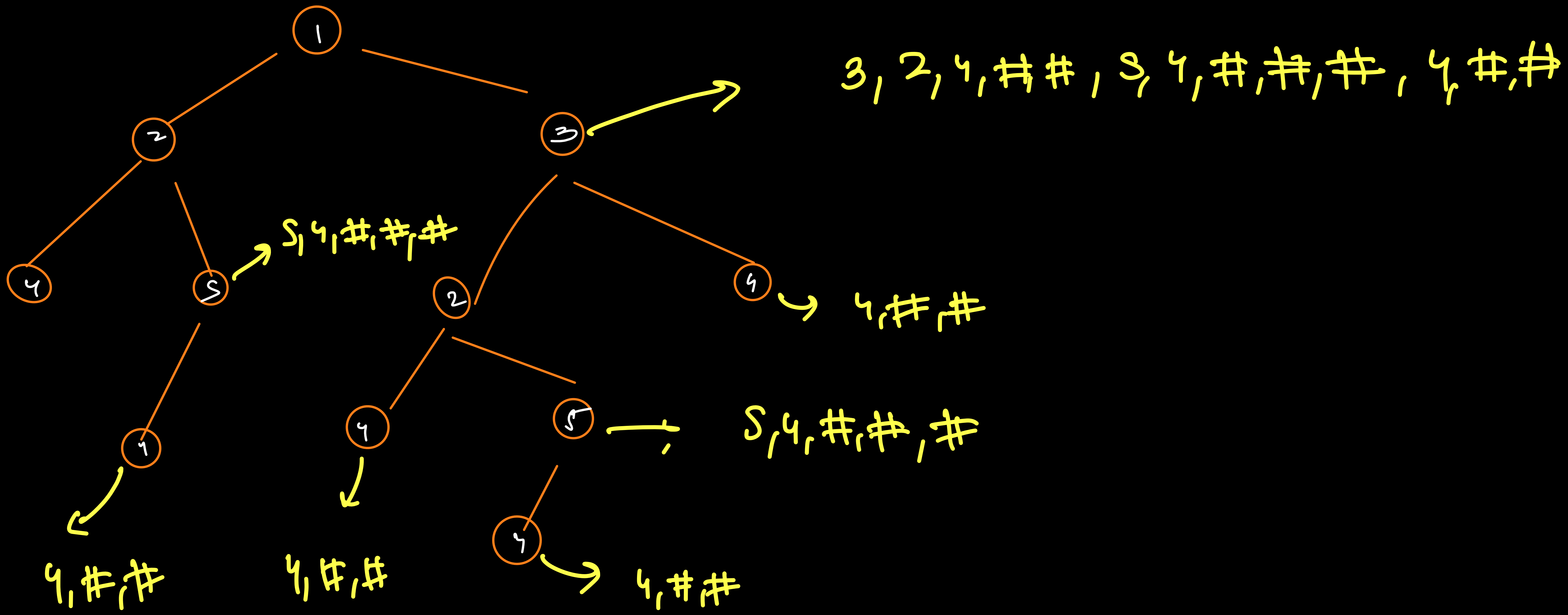
}

preorder of same subtree

JOIN THE DARKSIDE



JOIN THE DARKSIDE



JOIN THE DARKSIDE

f(root) = left= f(root.left)

right = f(root.riget)

String= root.val +","+ left +","+ right;

mp[string]++;

return Sry;

if(root==null) return "#";

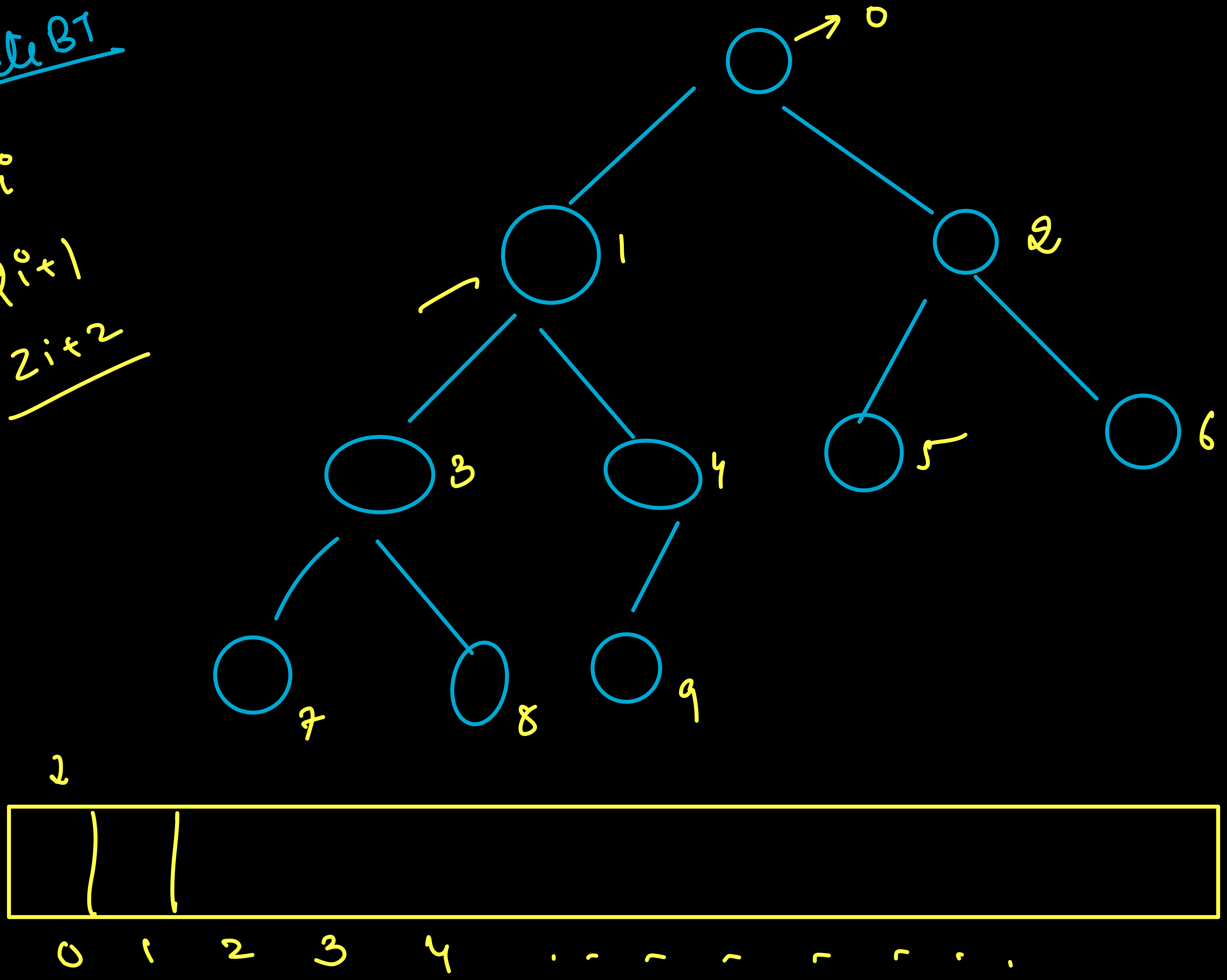
JOIN THE DARKSIDE

Complete BT

parent → i

left → 2i+1

right → 2i+2



JOIN THE DARKSIDE

Level order traversal

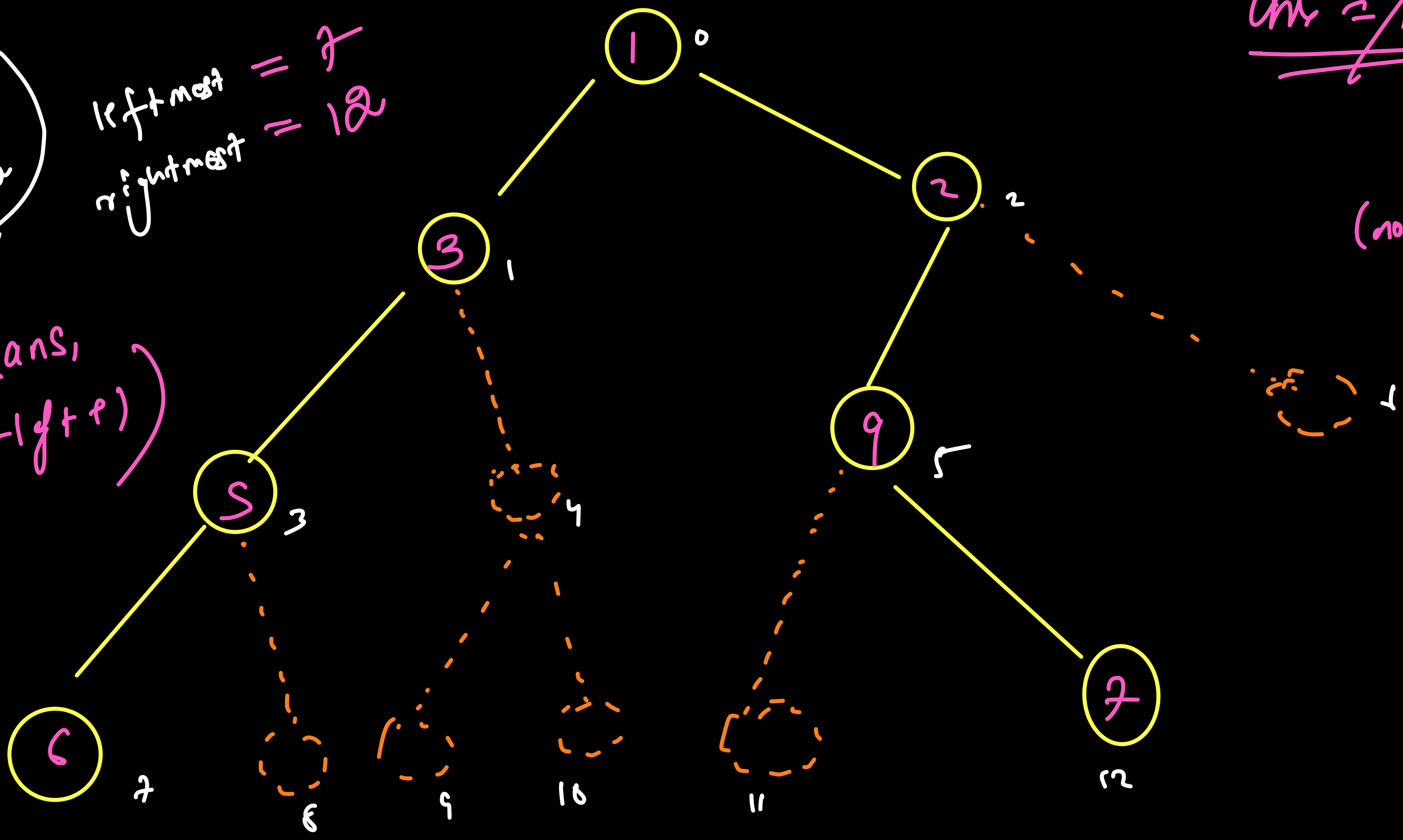
leftmost = 7

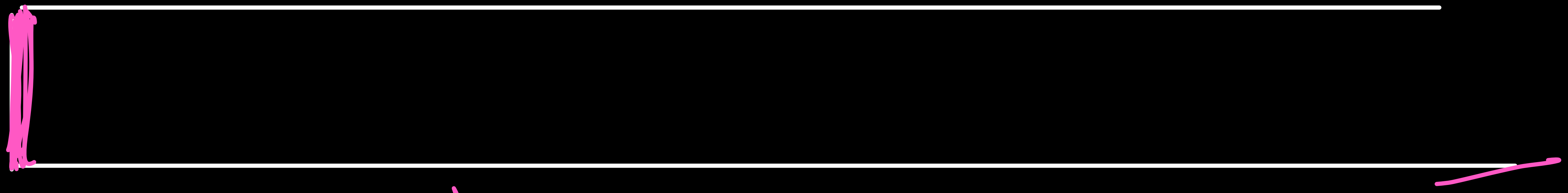
rightmost = 12

ans = max(ans, right-left+1)

ans = ~~1~~ ~~2~~ ~~3~~ 6

(node, idx)





(7, 12)

null

JOIN THE DARKSIDE

```
qu.enquee([root, 0])
qu.enquee (null)
left = null
right = null
last = null
while (not qu.isEmpty) {
    4 elemnt = qu.front();
     qu.dequee;
     if (elent [0] == null) {
          right = last [1];
```

JOIN THE DARKSIDE

```
                if( qu.isEmpty ()){
                    ans = max (ans, right - left +1)
                    break;
                } else {
                  qu.enqueue (null);

                }


       } else {
            if (last == null)
                 left = element [1]



       }
```

JOIN THE DARKSIDE