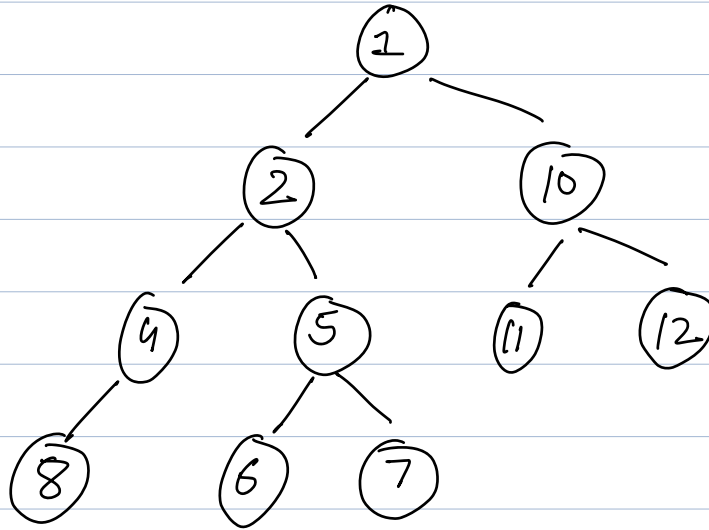
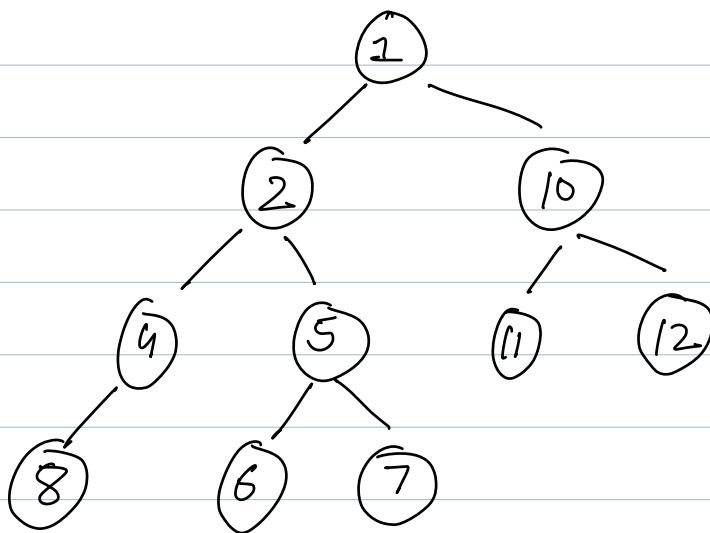


Iterative Inorder



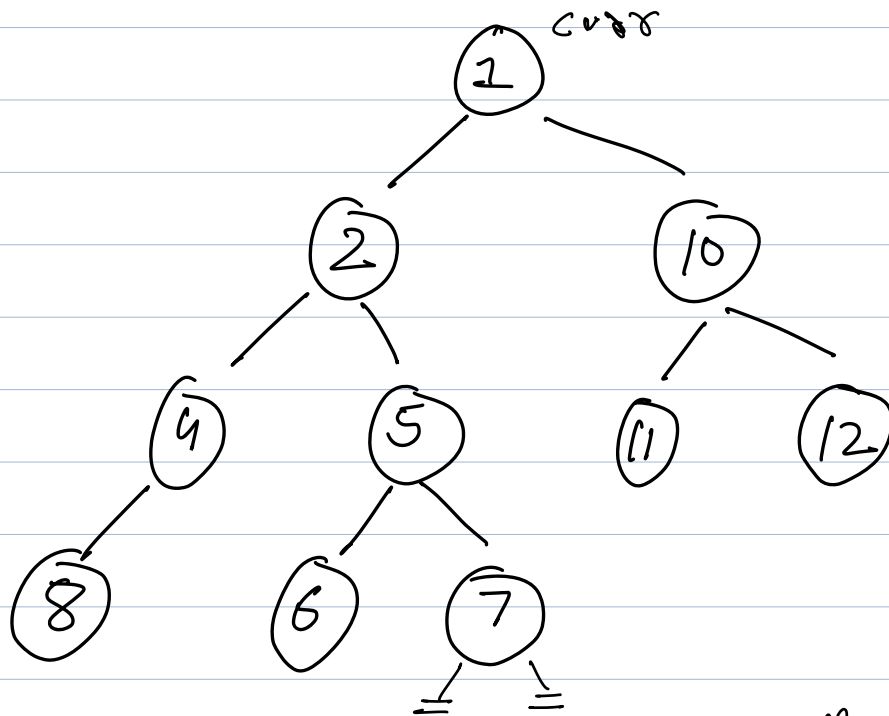
In order : 8 4 2 6 5 7 1 11 10 12

Tc: $O(n)$
Sc: $O(h)$



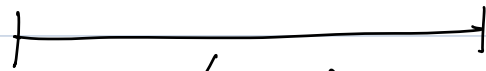
Inorder : 8, 4, 2, 6, 5, 7, 1, 11, 10, 12

Morris Inorder Traversal



→ will never have a right child.

⑦ is the inorder predecessor of ①



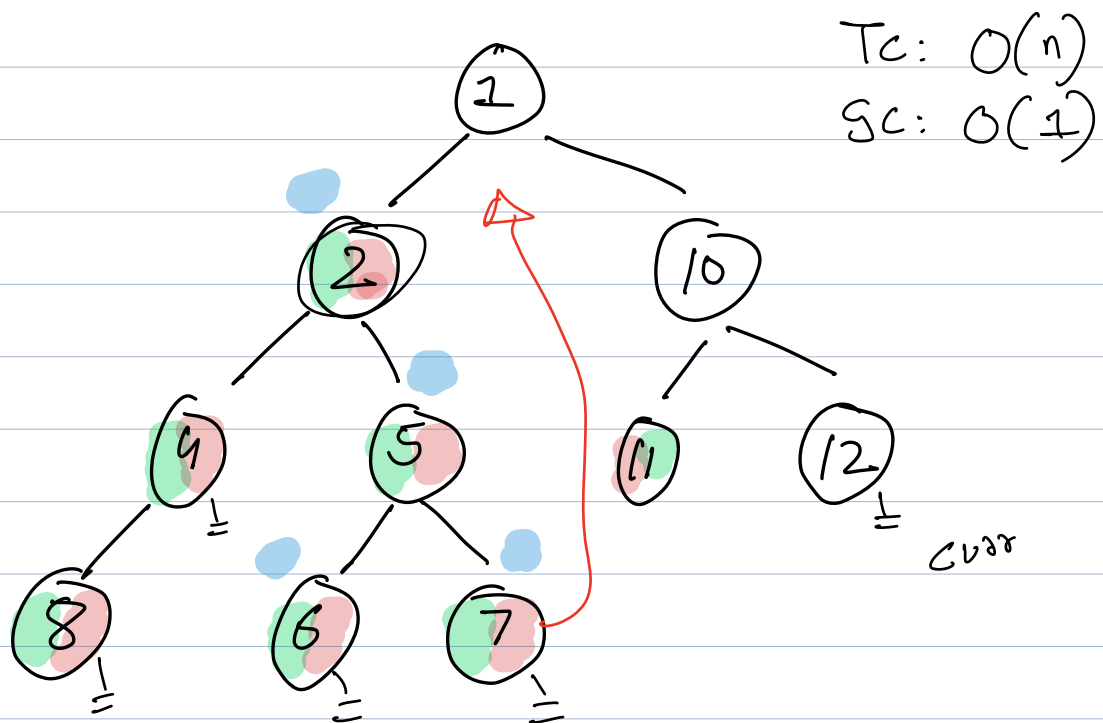
↪ if it exists in the LST

→ To find inorder predecessor.

1) are → curr.left



move to right till
there is no right child!



Inorder \Rightarrow 8 4 2 6 5 7 11 10 12

```
void MorrisInorder (Node root) {
```

```
    Node curr  $\Rightarrow$  root;
```

```
    while (curr != null) {
```

```
        if (curr.left == null) {
```

```
            print (curr.val);
```

```
            curr  $\Rightarrow$  curr.right;
```

```
        } else {
```

```
            Node temp  $\Rightarrow$  curr.left;
```

```
            while (temp.right != null && temp.right != curr) {
```

```
                temp = temp.right;
```

```
            }
```

```
            if (temp.right == null) {
```

```
                temp.right = curr;
```

```
                curr = curr.left;
```

```
            } else {
```

```
                print (curr.val);
```

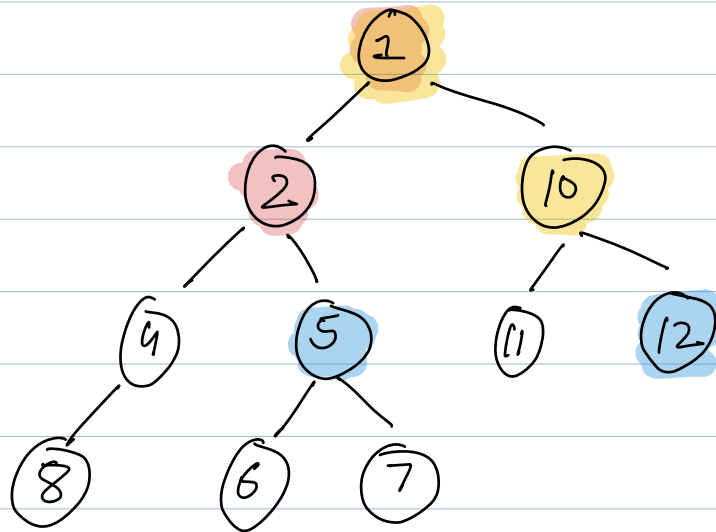
```
                temp.right = null;
```

```
                curr = curr.right;
```

```
            }
```

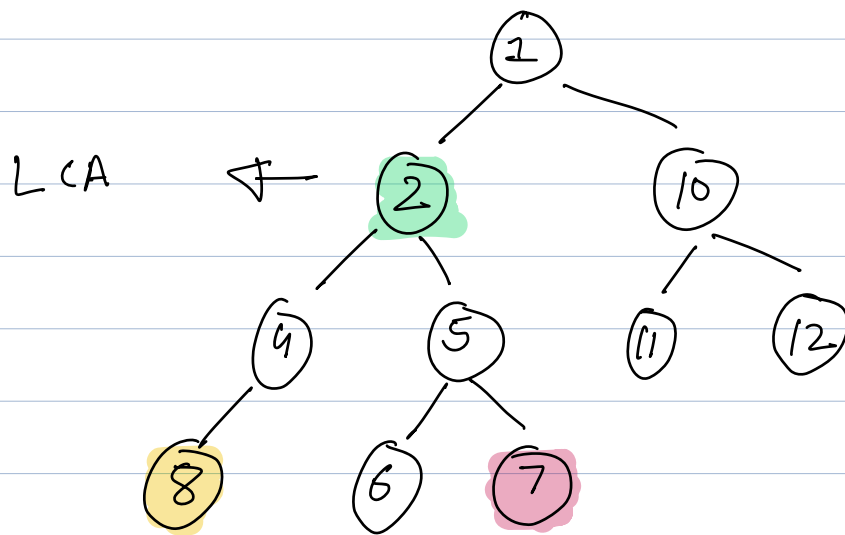
1 3 3

LCA [Lowest Common Ancestor]



Ancestors (5) \Rightarrow [1, 2, 5]

Ancestors (12) \Rightarrow [1, 10, 12]

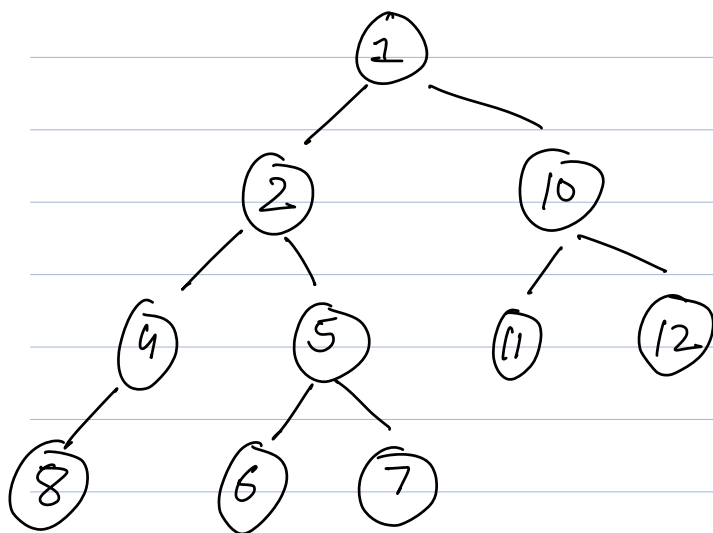


$T_C: O(n)$
 $SC: O(H)$

LCA (8, 7)

Ancestors (8) = 1 2 4 8
 Ancestors (7) = 1 2 5 7

LCA (8, 7) \Rightarrow 2

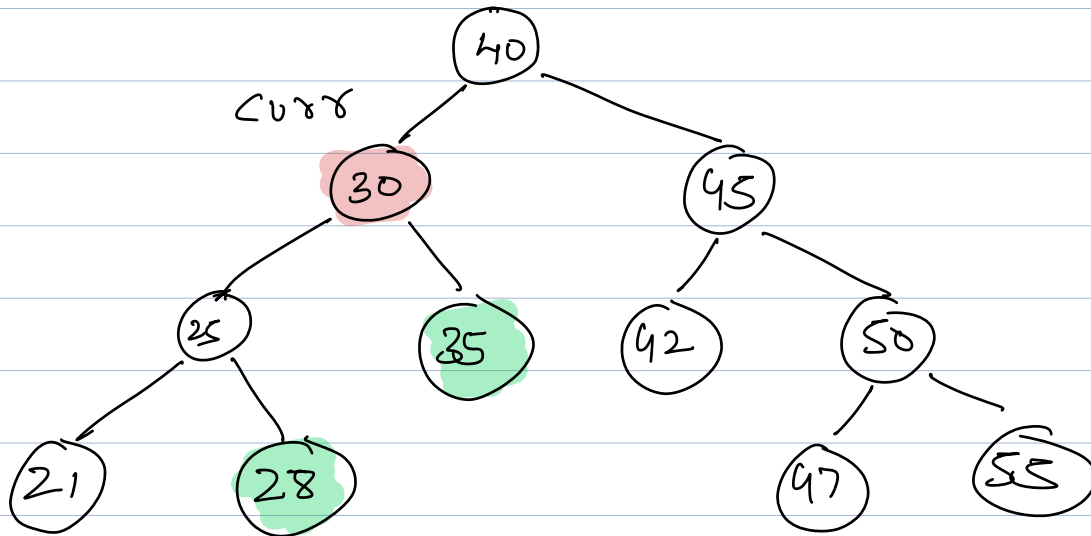


LCA (11, 12) \Rightarrow 10

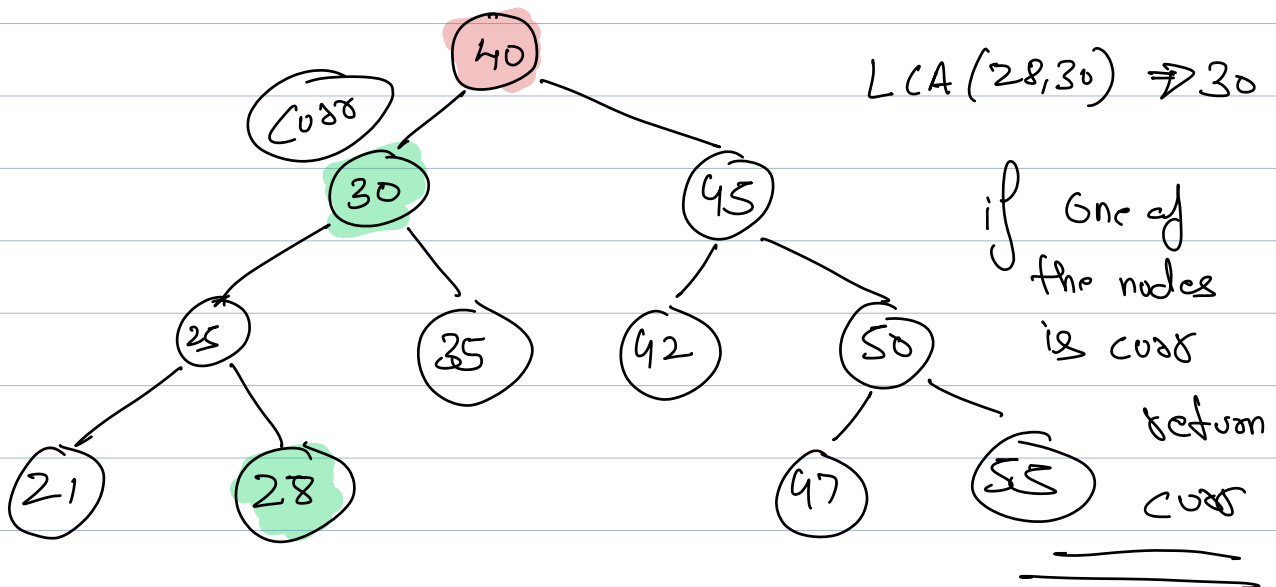
LCA (4, 2) \Rightarrow 2

Ancestors (4) \Rightarrow 1, 2, 4
 Ancestors (2) \Rightarrow 1, 2

LCA in a BST

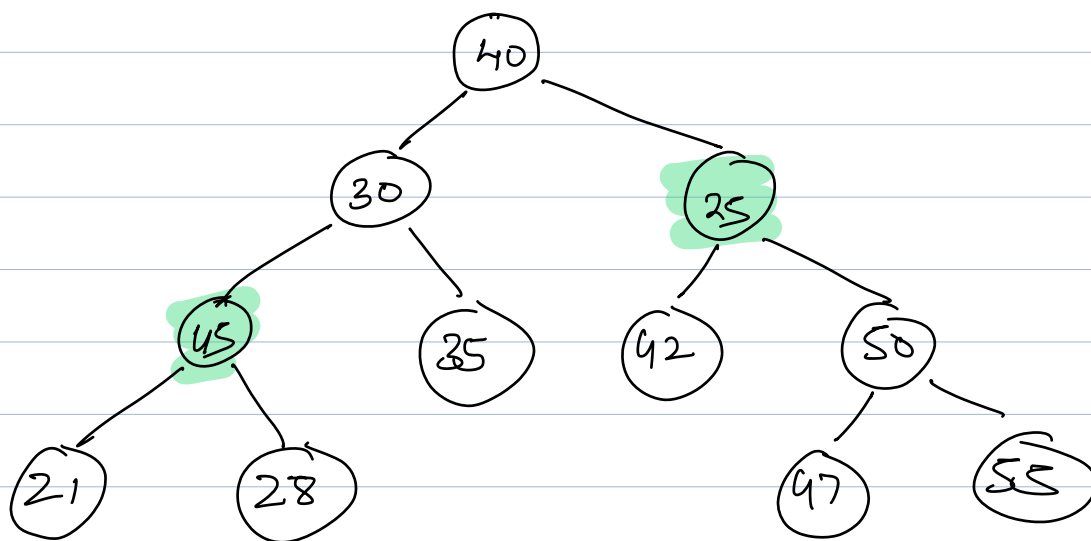
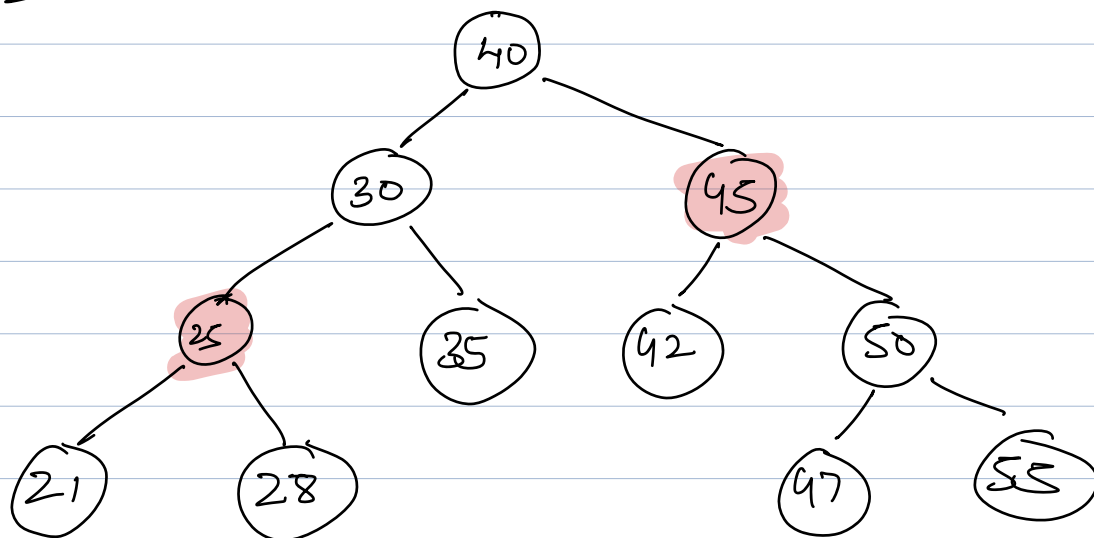


$$LCA(28, 35) \Rightarrow 30$$



$$Tc: O(H) \quad Sc: O(1)$$

①



21, 45, 28, 30, 35, 40, 42, 25, 47, 50, 55

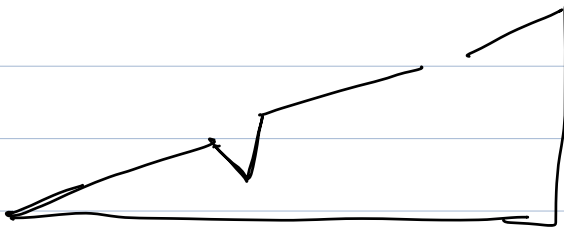
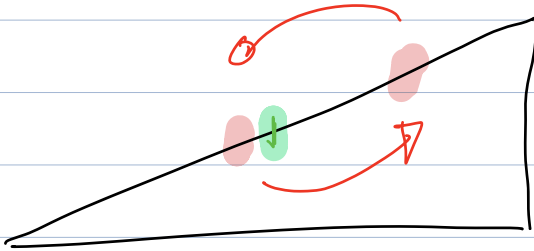
$arr[i] > arr[i+1]$
n ↑ ↑

$arr[i] > arr[i+1]$
 ↑

first

second

second



1 2 3 4 5 6.

1 2 3 5 4 6

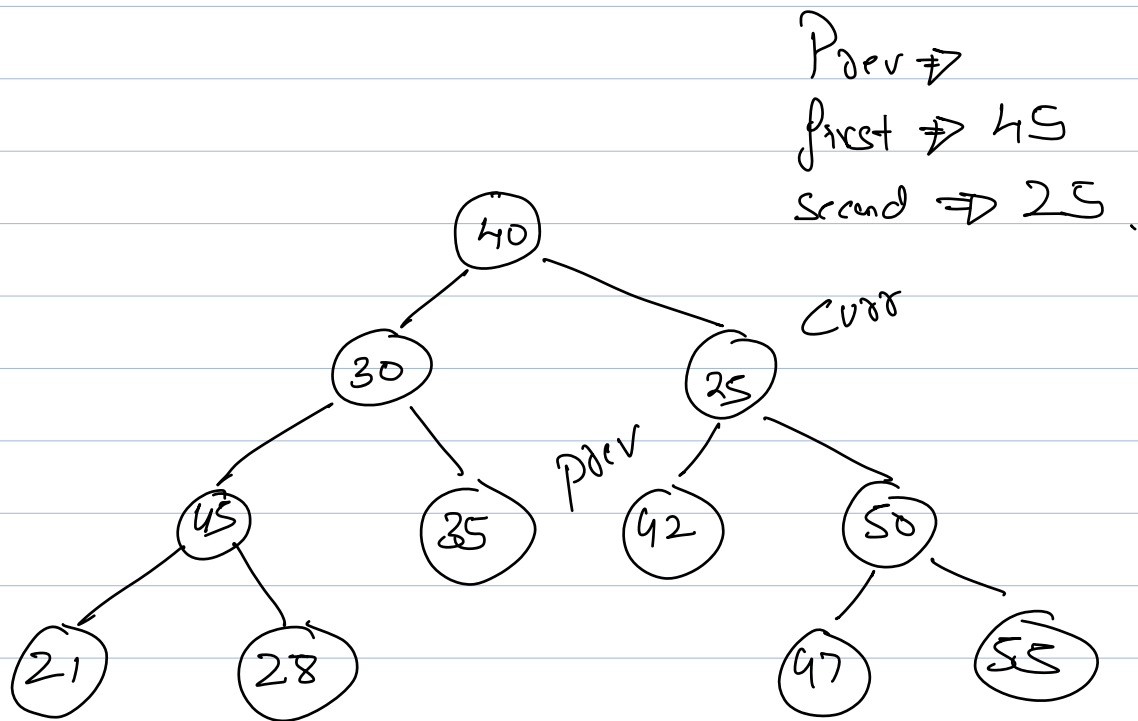
$arr[i] > arr[i+1]$

Tc: $O(n)$

Sc: $O(n)$

Recursive Inorder: $Sc: O(H)$

Inorder Morris: $Sc: O(1)$



$Tc: O(n)$

$Sc: O(H)$

prev = null first = null second = null.

prev = null, first = null, second = null;

curr = root;

void findCulprits (Node root) {

findCulprits (root.left)

if (prev != null && prev.val > curr.val) {

if (first == null) {

first = prev;

}

second = curr;

}

prev = root;

findCulprits (root.right);

}