

---

---

---

---

---



---

---

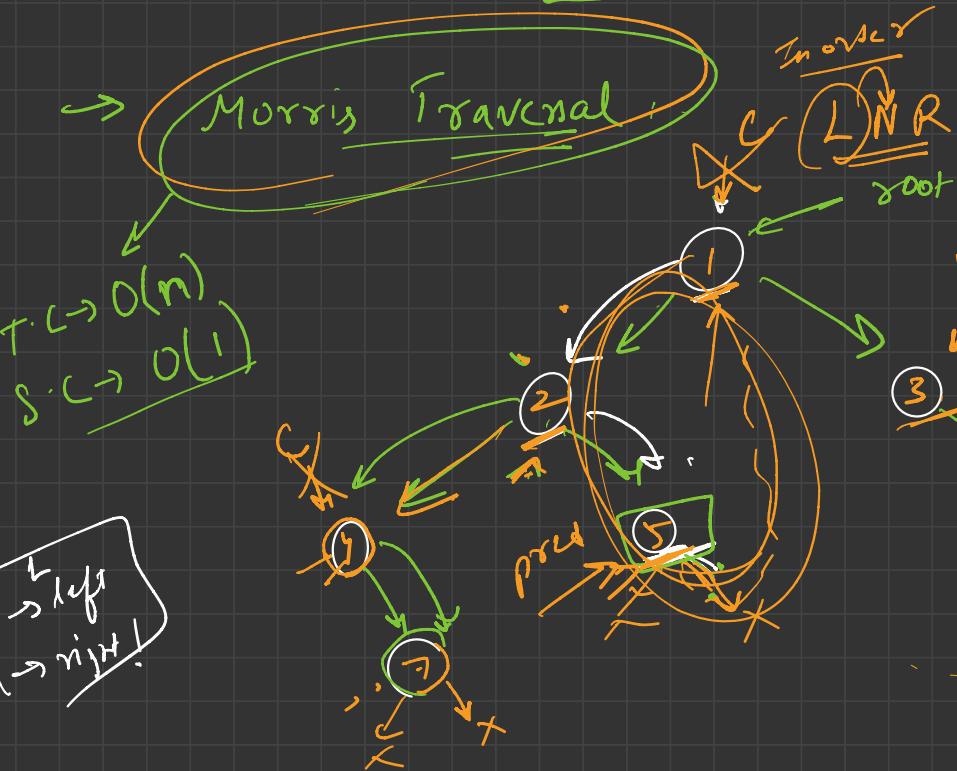
---

---

---



# Trees

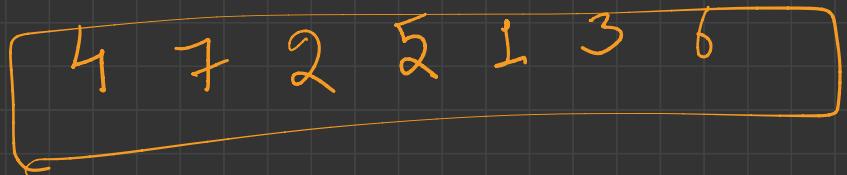


```

Algo:
    current = root
    while (curr != NULL)
        if left not exists
            visit (current)
            current = current -> right
        else
            pred = find (curr)
            if (pred -> right == NULL)
                pred -> right = current
                current = current -> left
            else
                pred -> right = NULL
                visit (current)
                current = current -> right
    }
}

```

temp link removed



Inorder

prev = curr  $\rightarrow$  left

T.C  $\rightarrow O(n)$

S.C  $\rightarrow O(1)$

while ( $\underline{\text{prev} \rightarrow \text{right}} \neq \text{NULL}$ )  ~~$\text{prev} \rightarrow \text{right} = \text{current}$~~

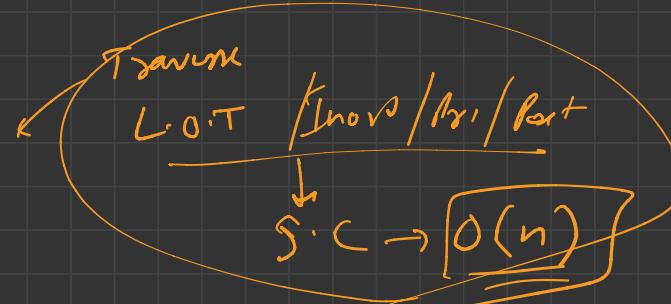
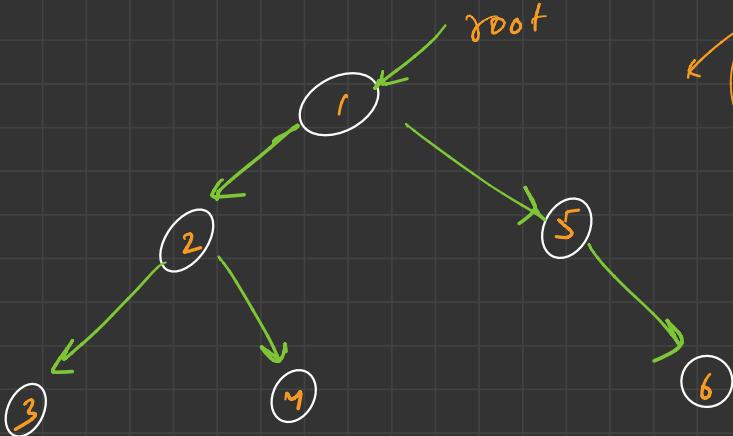
↓

\_\_\_\_\_

right

3

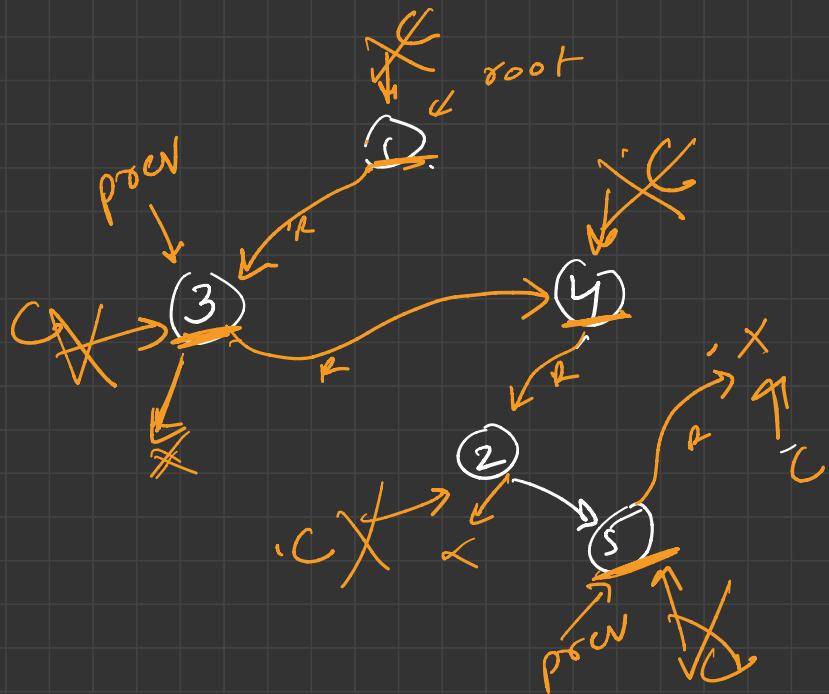
→ flatten a tree into LL



# 1 → traverse → root node create  
K & Juge  $\times$  → in-place

# 2 → recursion →  $O(n)$   $O(n)$   $\rightarrow \times$

#3 → Morris traversal



Algo:-

→ current = root

while (curr != NULL)

{

if curr left exists

{

    prev = curr → left  
    while (prev → right)  
        prev → right;

prev → right = curr → right

curr → right = curr → left

}

curr = curr → right  
}

# Morris traversal

PreOrder  $\rightarrow$   $\frac{N}{\downarrow} \frac{L}{\uparrow} \frac{R}{\uparrow}$   
Root

