

---

---

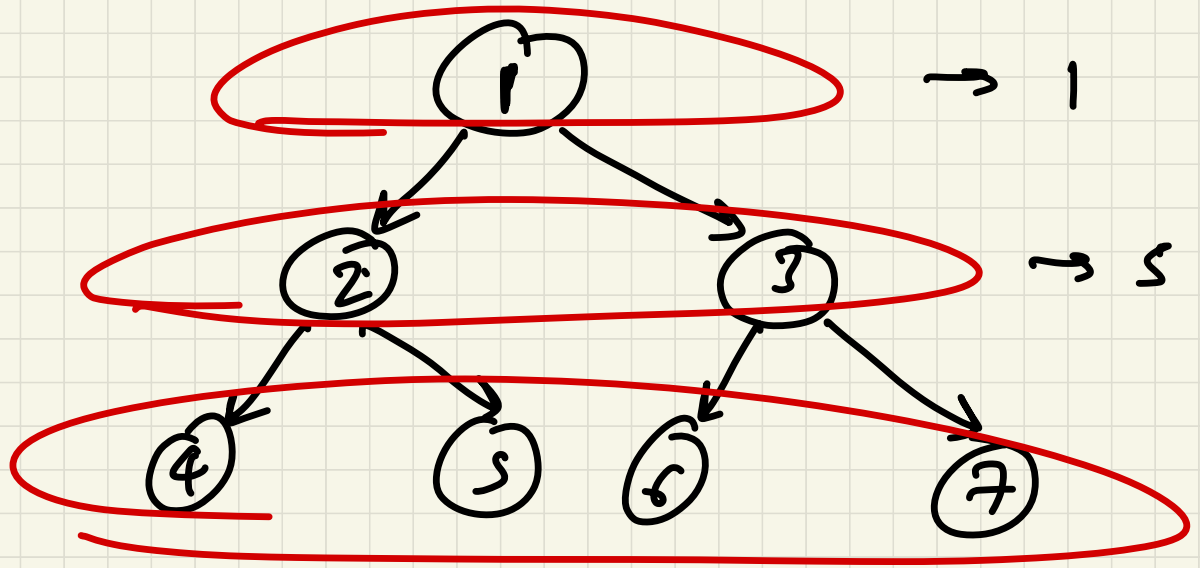
---

---

---



# ① BFS



1 → 2 → 3 → 4 → 5 → 6 → 7

When to use :

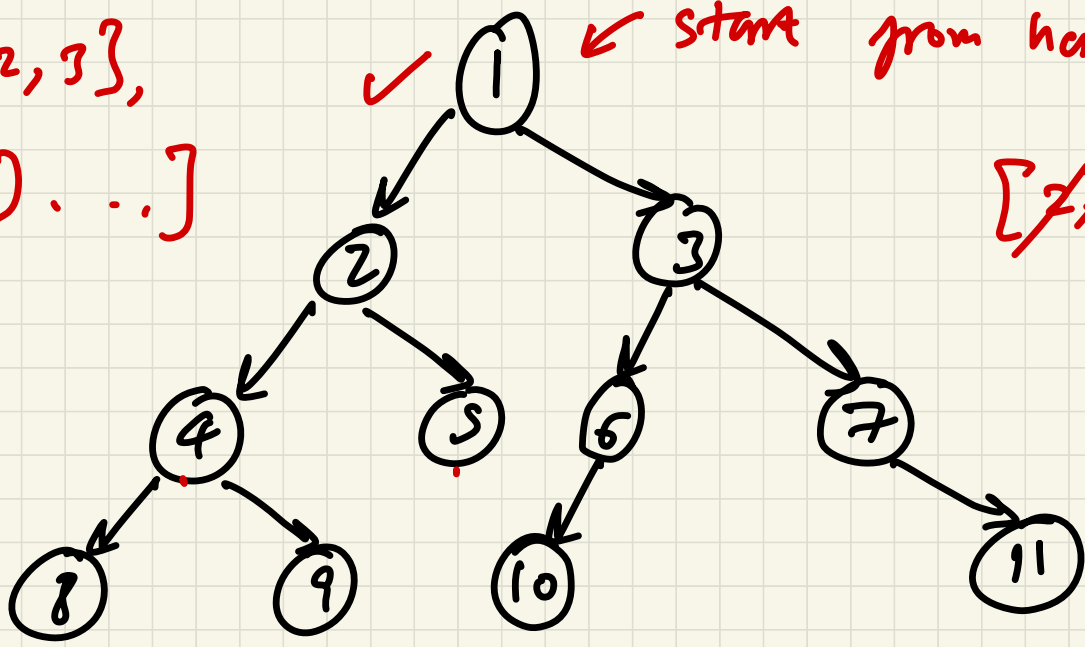
- ① when answers lies near the root node.
- ② when you are asked to search by level.

[ [1], [2, 3],  
[4, 5, 6, 7] ... ]

✓ 1 ← start from here

Queue

~~[7, 8, 9, 5,~~  
~~6, 7]~~  
~~8, 9, 10,~~  
..



[X, 2, 3  
4, 5, 6, 7]

1 → 2 → 3 → 4 → 5 → 6 → 7 → 8,  
9, 10, 11

✓✓

[1,

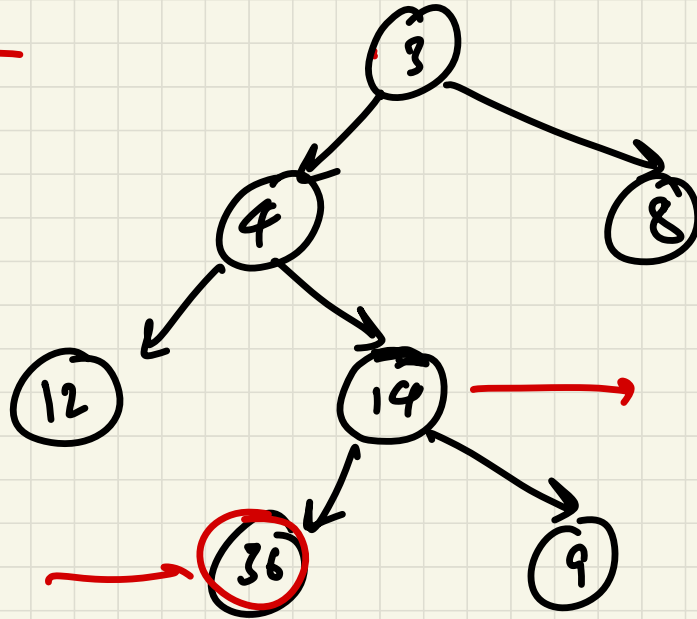
Google

[1, 2, 3, 4, 5  
6, 7]

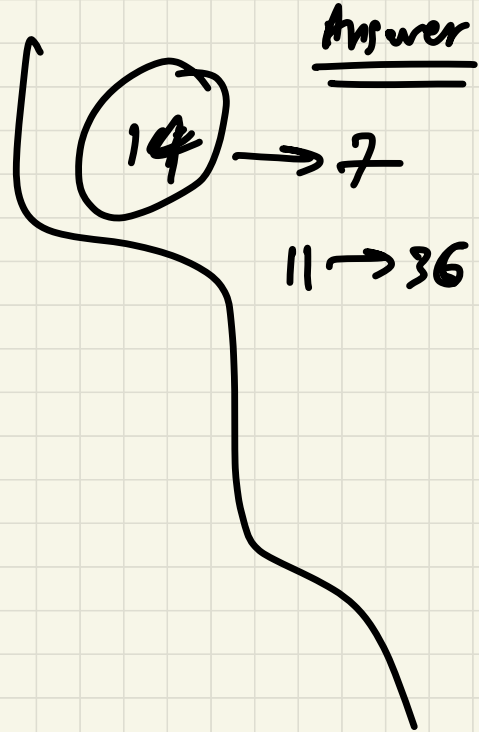
$$l_s = 2$$

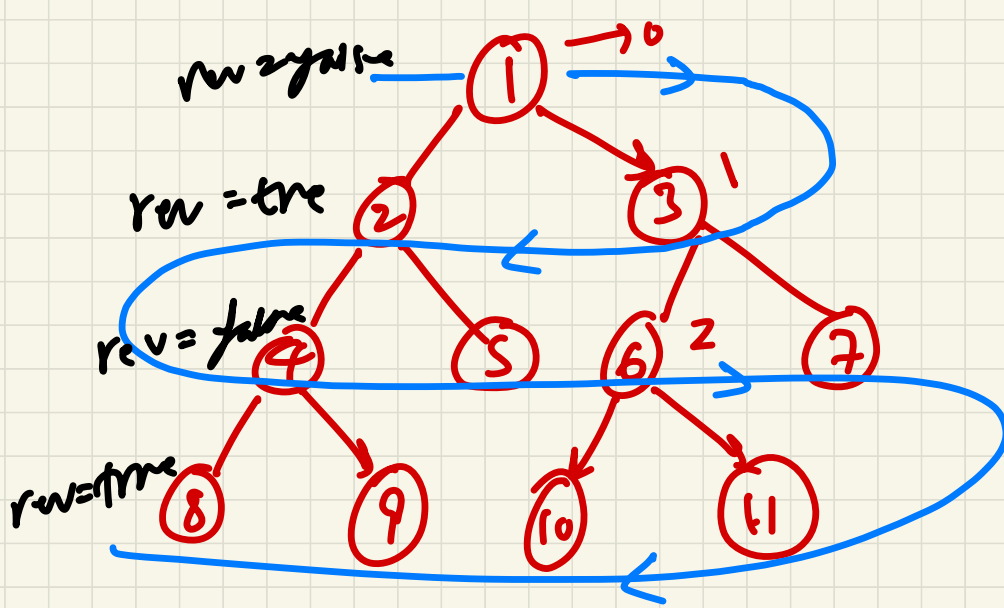
$$a_l = \frac{2 + 3}{2} = 2.5$$

Google:



[~~3~~, ~~4~~, ~~8~~, ~~12~~, ~~14~~, ~~36~~, 9]





1 4, 5 ~~3~~  
 3, 2  
 4, 5, 6, 7  
 11, 10, 9, 8

1  
 3, 2  
 4, 5, 6, 7

✱ Normal order

→ remove from front

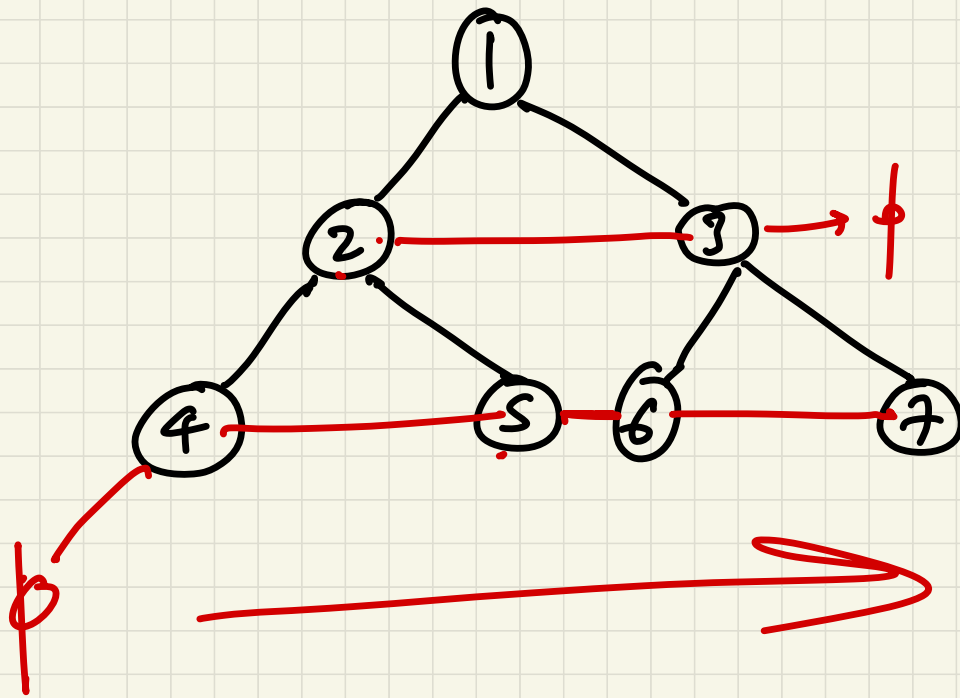
→ add in back

✱ Rev → remove back  
 add front

~~1, 2, 3, 4, 5, 6, 7~~, 8, 9, 10, 11

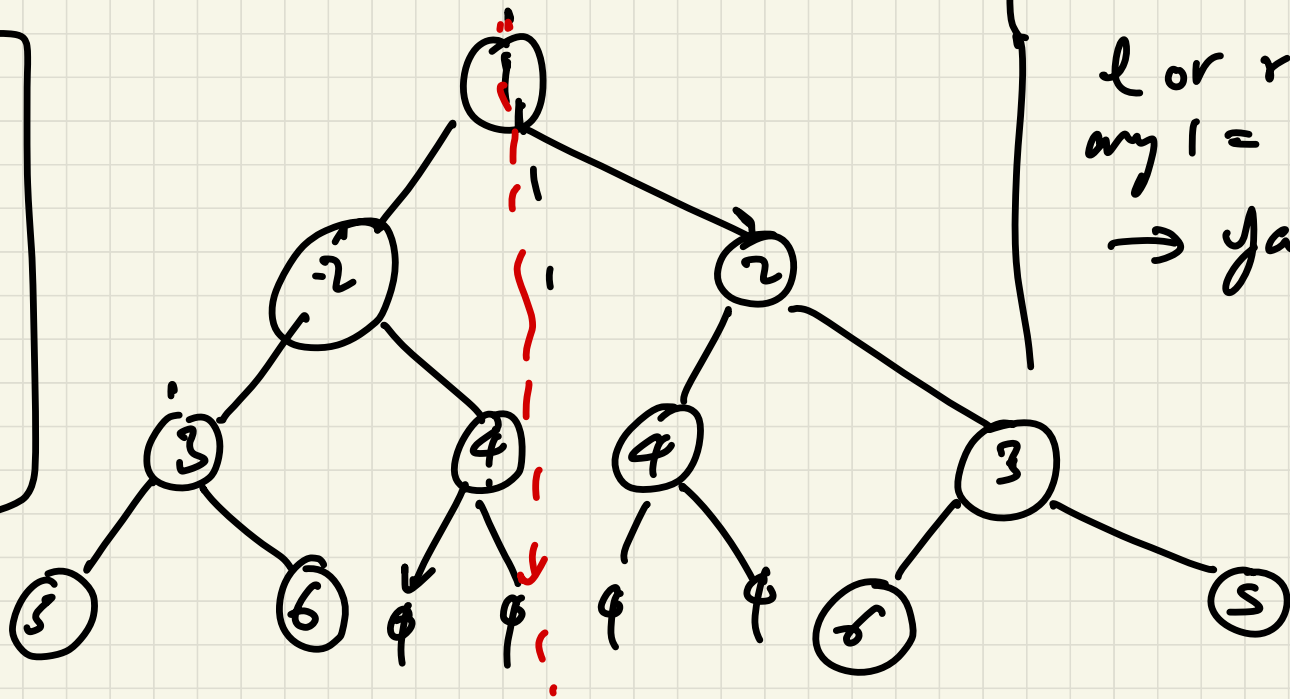
deque

6, 7, ~~8~~



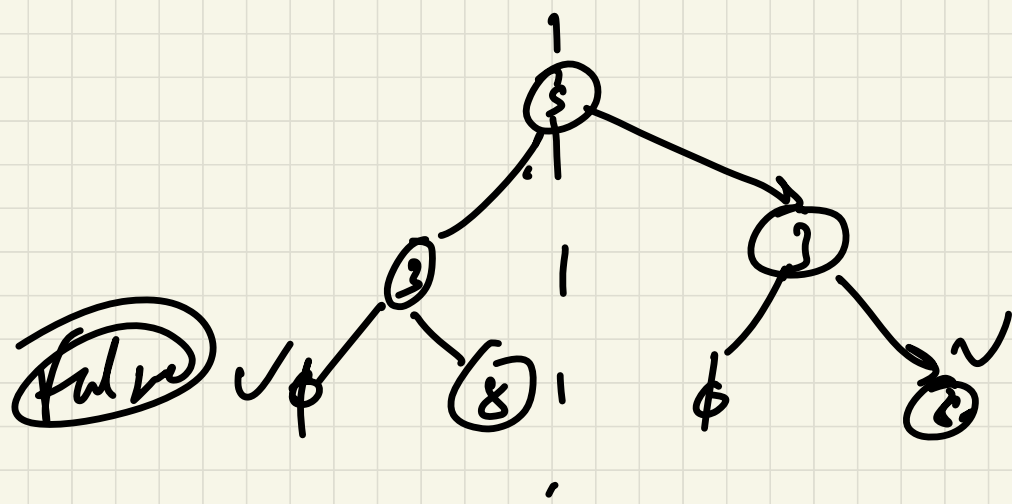
left most	current
+	+
<del>2</del>	<del>2</del>
4	3

l.l  
r.r  
l.v  
r.l

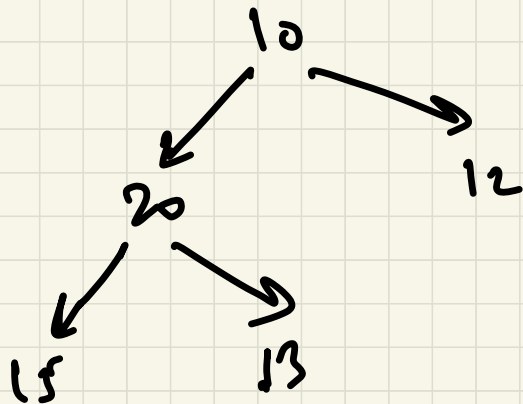
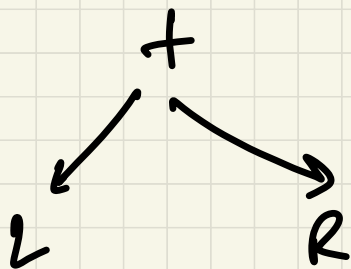


l or r  
any l = r  
→ false

[7, 2, 3, 5, 4, 4, 5, 5, 6, 6]







Pre order

$+ \rightarrow L \rightarrow R$

In-order

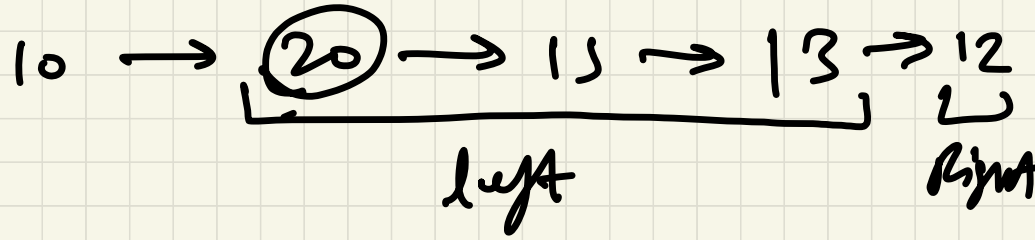
$L \rightarrow + \rightarrow R$

Post-order

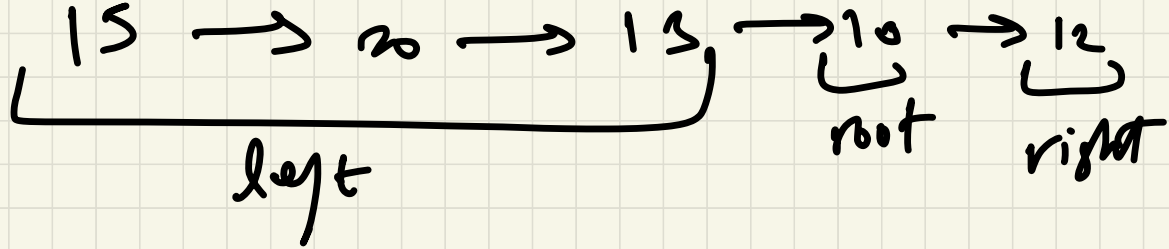
$L \rightarrow R \rightarrow +$

pre-order

Root  $\rightarrow$  Left  $\rightarrow$  Right



In order



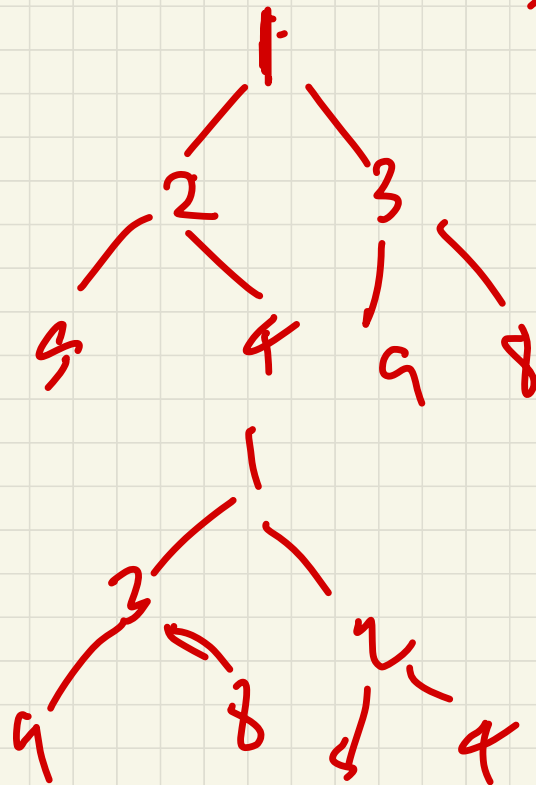
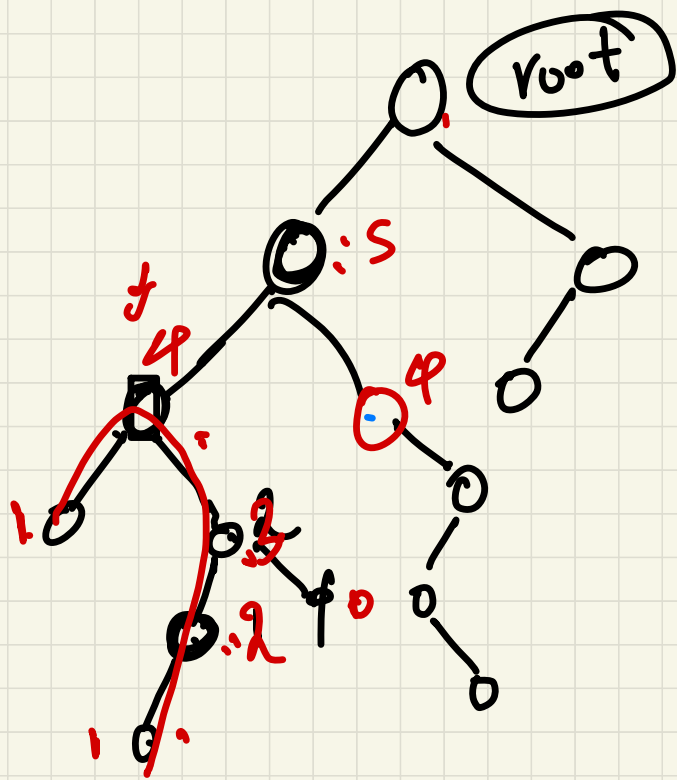
left most first:

In BST  $\rightarrow$  when  
you want to visit  
value

Post order : 15, 13, 20, 12, 10  
                                left                          right                          root

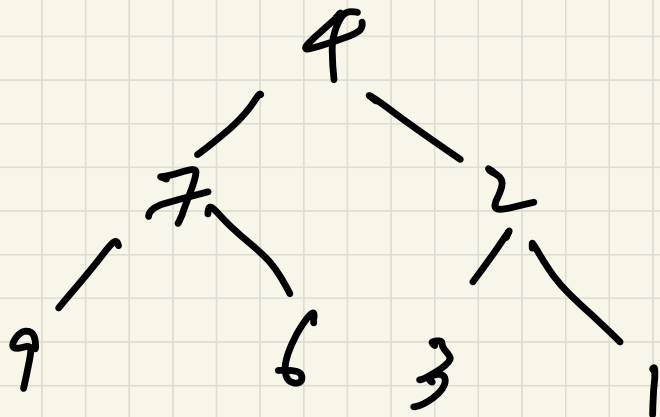
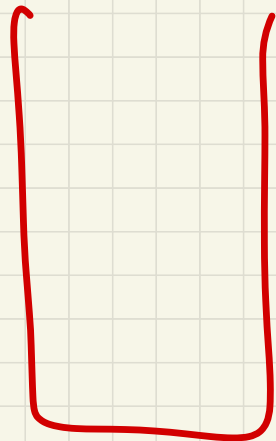
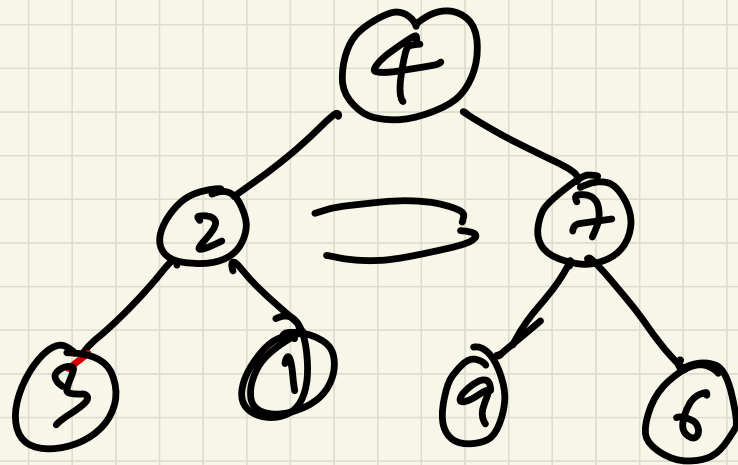
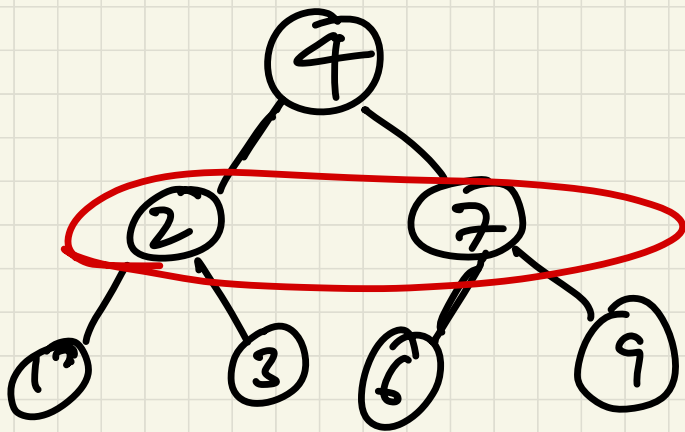
(when you delete a tree)

---

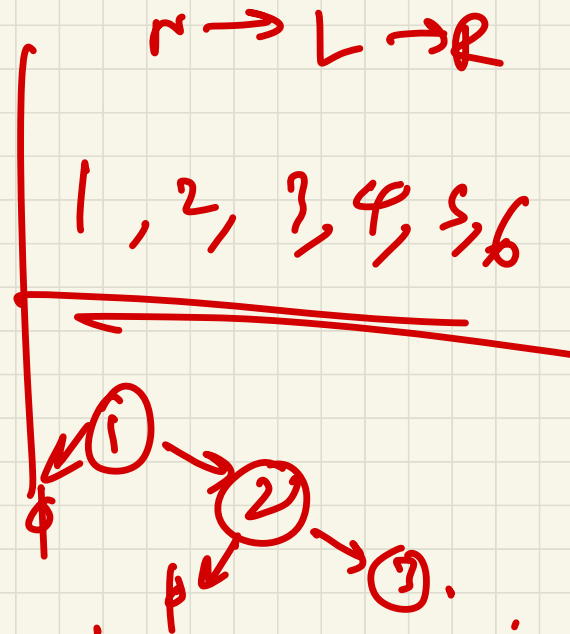
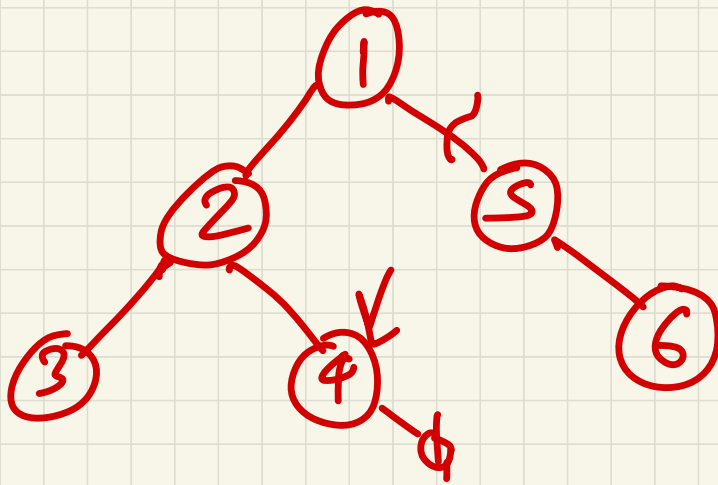


dia

$$\begin{array}{r} \cancel{0} \quad \cancel{3} \quad 4 \\ - 1 \\ \hline = 3 \end{array}$$



# Facebook:



① Normal pre order traversal

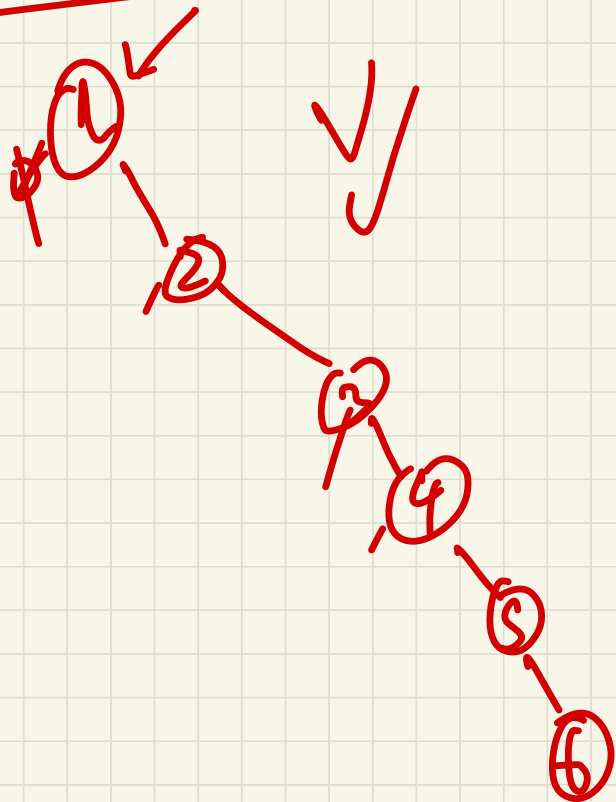
② As you traverse, store nodes in queue.

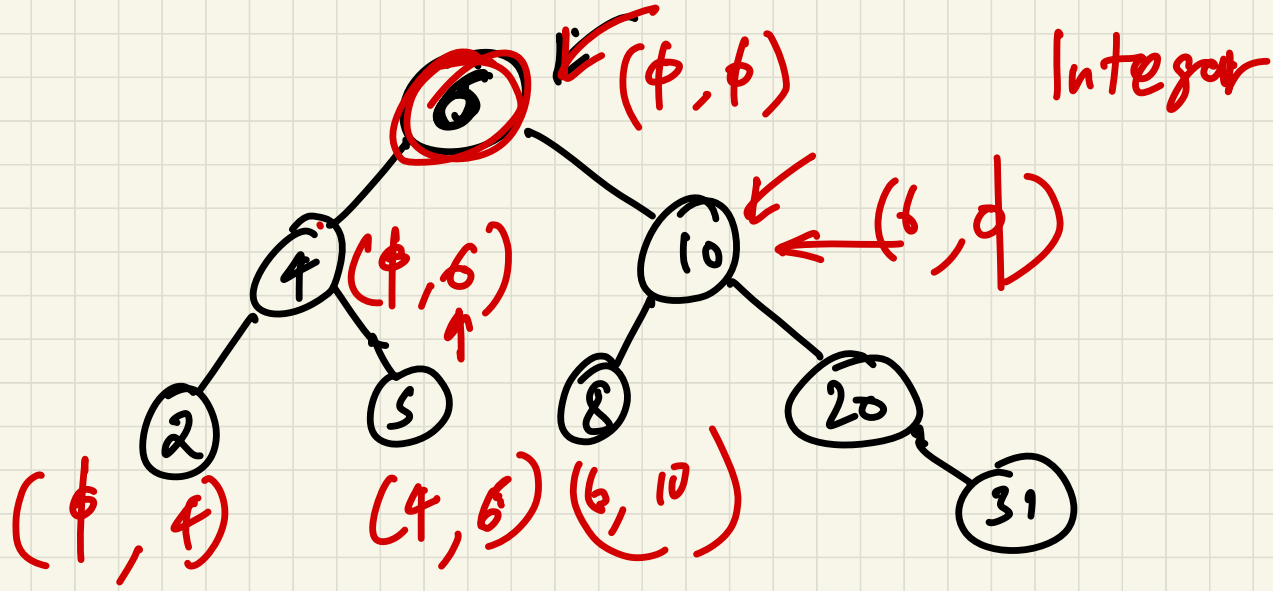
③ In the end, remove from queue & make linked list.

$$\underline{S \& T \Rightarrow O(N)}$$

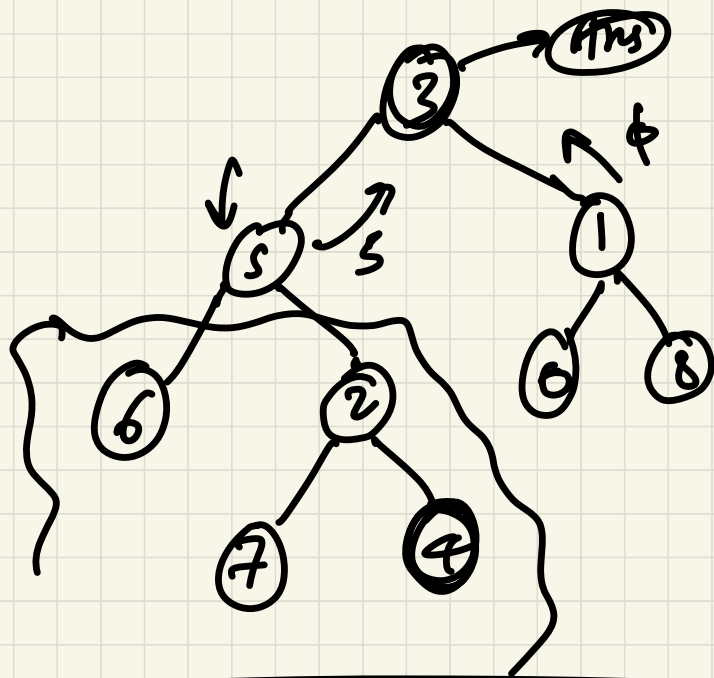
$O(1)$  space?

$N \rightarrow L \rightarrow R$









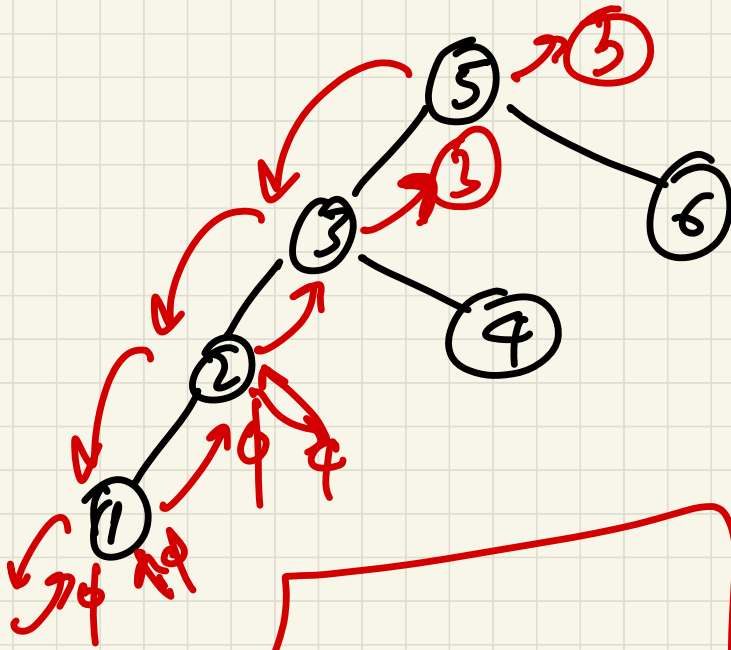
$p$   $q$   
 $(5, 4)$

if (node =  $p$  or  $q$ )  
 return node;

left  
 right

if (left & right !=  $\phi$ )  
 return node

if left =  $\phi$ , right = ans  
 else left = ans



$k=3$  |  $\text{count} = 0$   
 $+ 1$   
 $\cancel{2}$   
 $3$

left  
 if ( $\text{left} \neq \emptyset$ )  
     return left

count ++  
 if (count == k)  
     return node

right

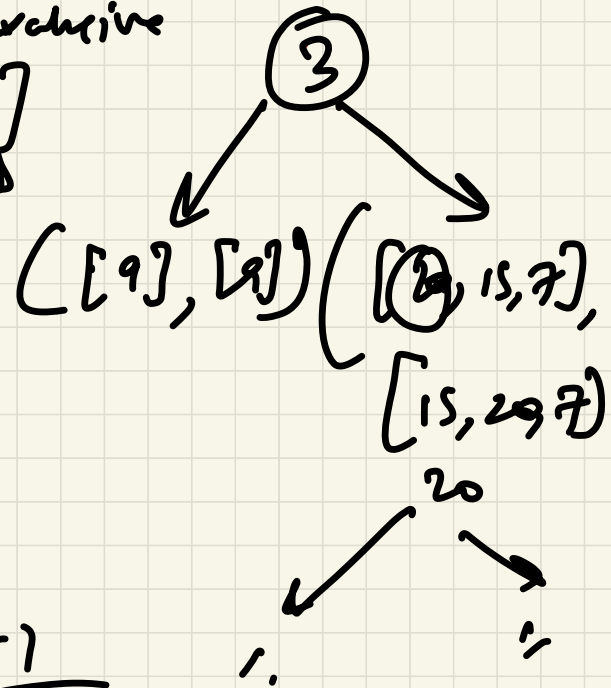
pre = [3, 8, 9, 20, 15, 7]

index = 1, 3

exclusive

0 1 2 3 4 5

right



inorder = [8, 9, 3, 15, 20, 7]

index

left

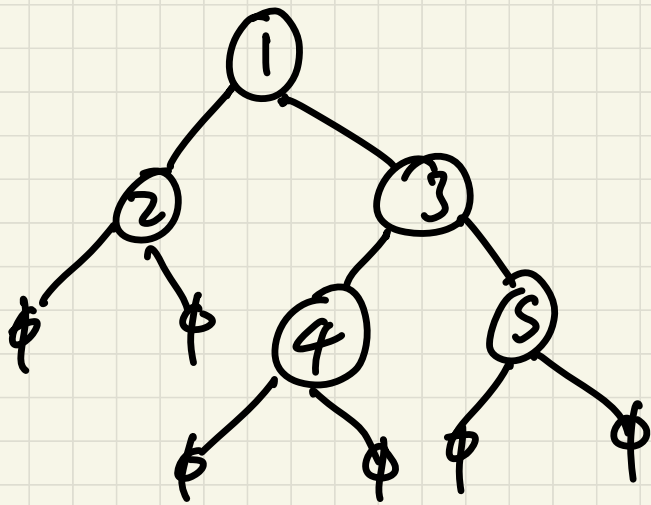
right

0 1 2 3 4 5

→ index + 1

l = (1, index + 1)

r = (index + 1, end)

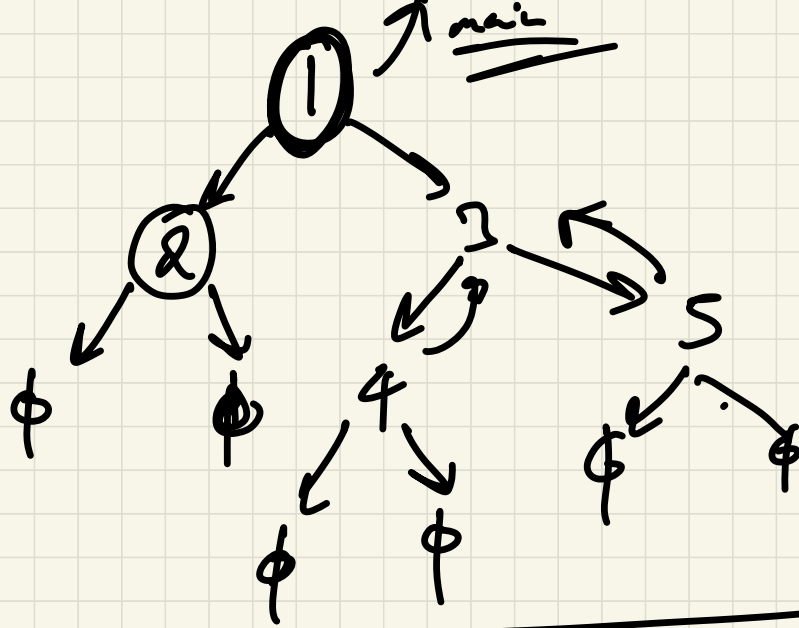


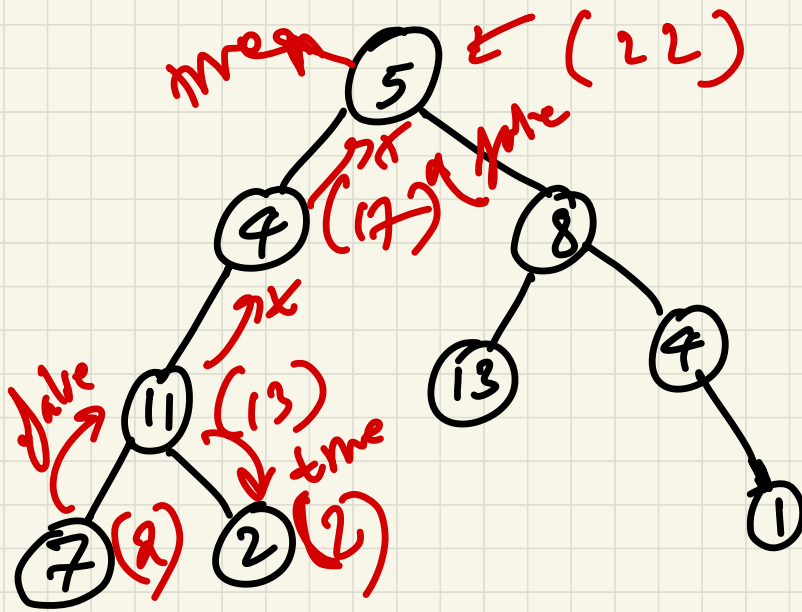
$\nearrow$  root  
 $[2, 1, 4, 3, 5]$

① have pre & in order stored.

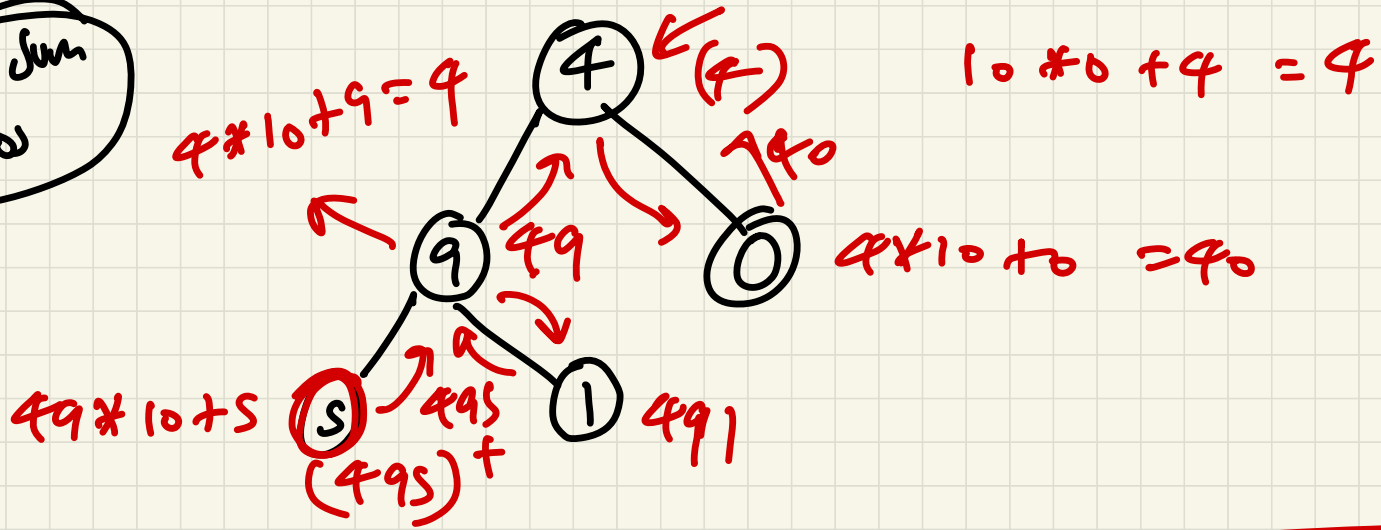
② store pre order with  $\phi$ .

$\langle \underline{1}, \underline{2}, \underline{\phi}, \underline{\phi}, \underline{3}, \underline{4}, \underline{\phi}, \underline{\phi}, \underline{5}, \underline{\phi}, \underline{\phi} \rangle$

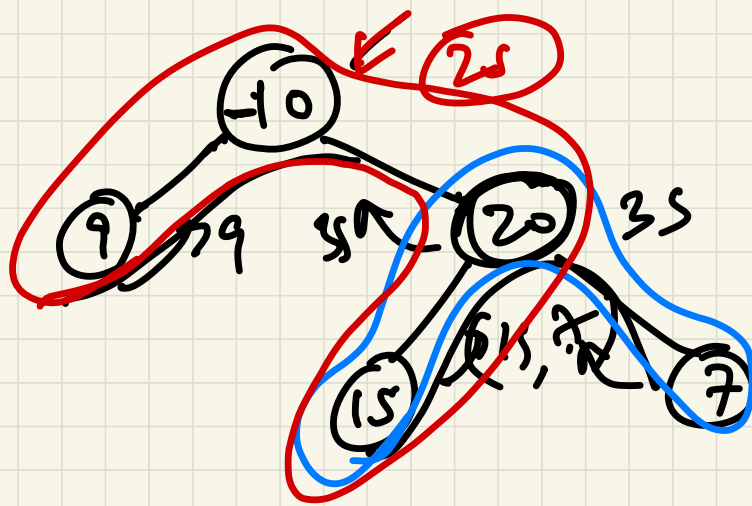




Path Sum  
no



global  
 $\frac{2}{9}$   
 42



left  
 right

current + left  
 + right

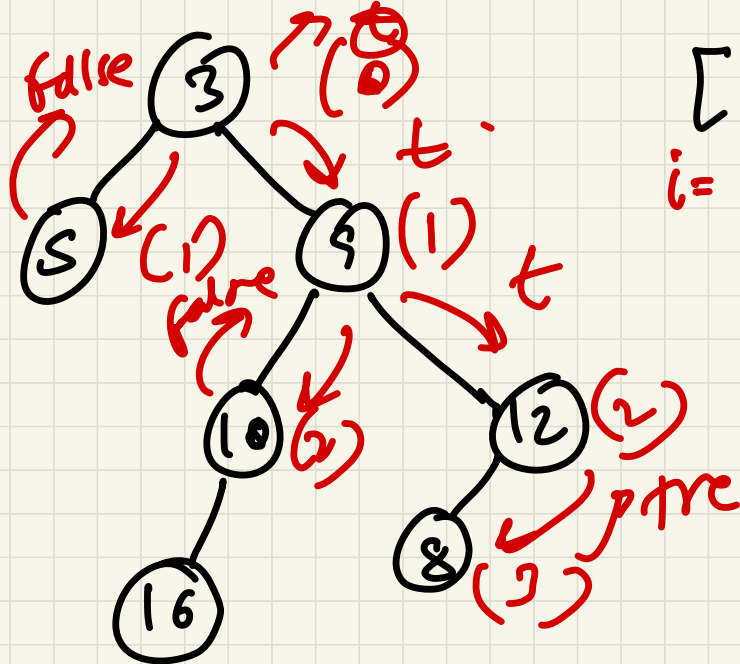
update global

path with  
 max sum

max ( left, right )  
 & current



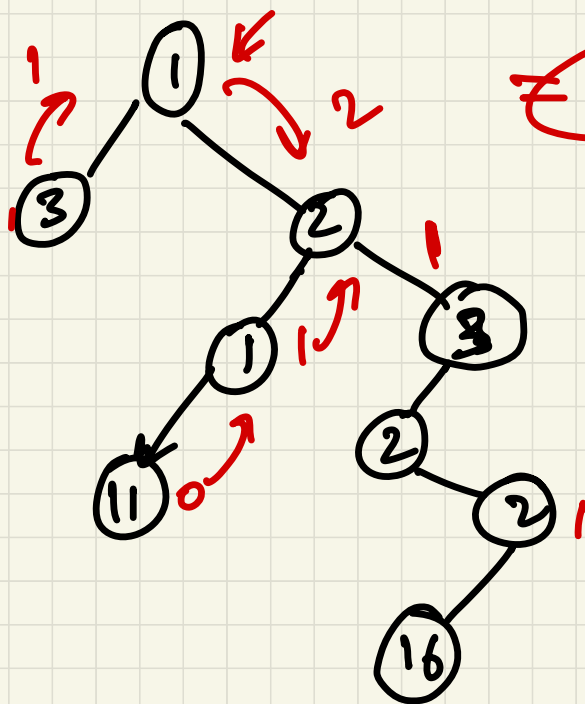
⑦



[<sup>↓</sup>3, 9, 12, 8]  
i = 0

②

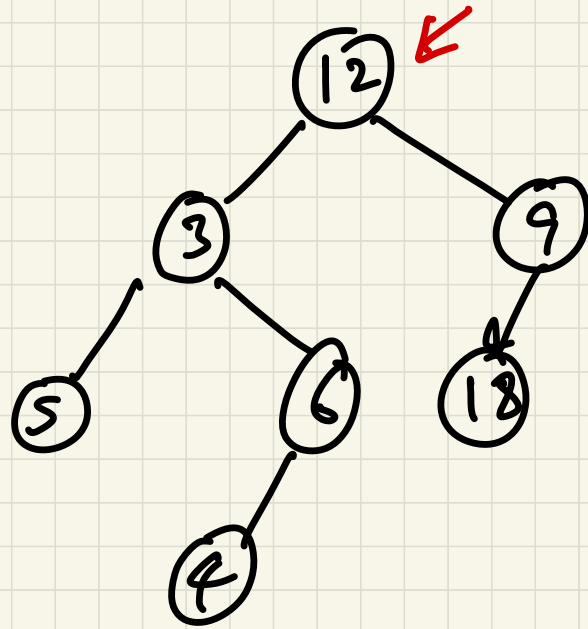
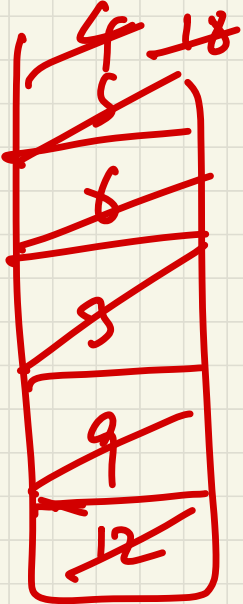
$\langle 1, 2, 3, 2, 2 \rangle$



③

④

## DFS using Stack



12, 3, 5, 6, 4, 9, 18

