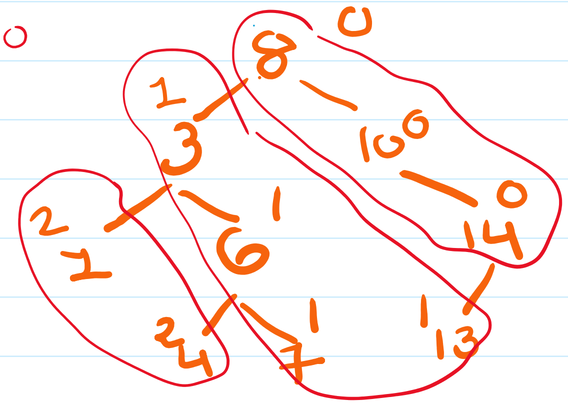
27 March 2024 18:41

gfg

Diagonal Traversen

d = 0

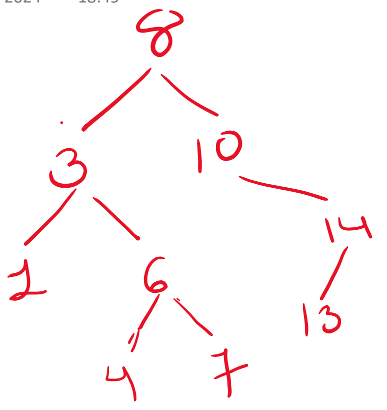


=) going left +1
going right
noting

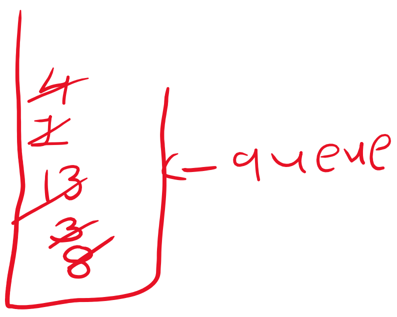
0 => 8 10 14
1 => 3 6 7 13
2 => 1 4

ans
{ 8 10 14 3 6 7 13 1 4 }

map using



8	10	14
3	6	7
13	1	4

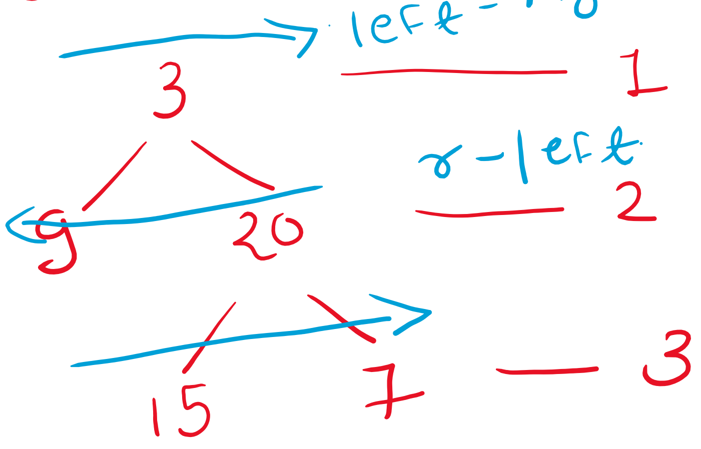


right print
left ele.
push

```
temp => 8
while(temp != null)
{
    print temp val
    if (temp -> left)
        q.push
    temp = temp -> right
```

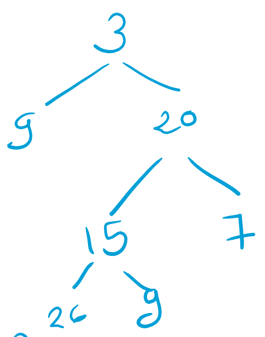
27 March 2024 19:03

LC :- 103 Zig-zag Traversal

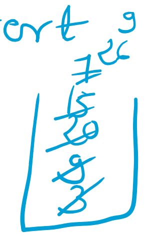


[3] [20 9]
[15 7]

27 March 2024 19:07



root insert



flag=true 3

flag=false 9 20 $\leftarrow$ L to R

3 print

$\uparrow$ child push

flag $\Rightarrow$ L to R = true

9

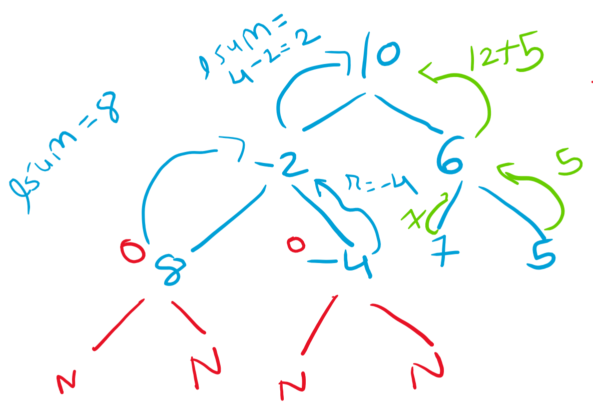
[20, 9] $\uparrow$ vector ની end માં push કરવાનું

[15, 7] start કરવાનું } flag false થાય ત્યારે

[9, 26]

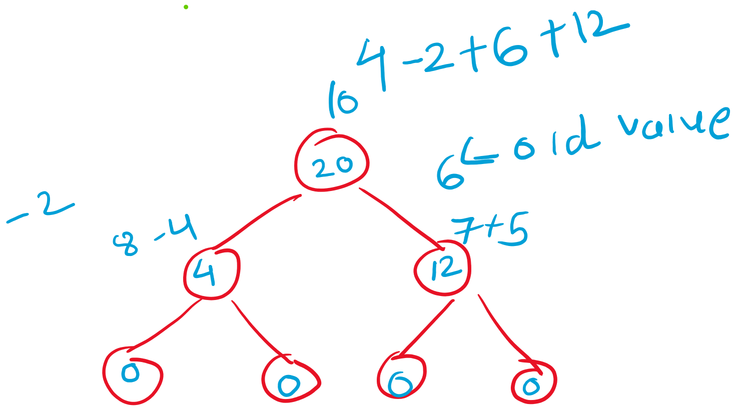
27 March 2024 19:25

gfg: Transform to sum Tree



(1) left Node will be zero

left, right value sum

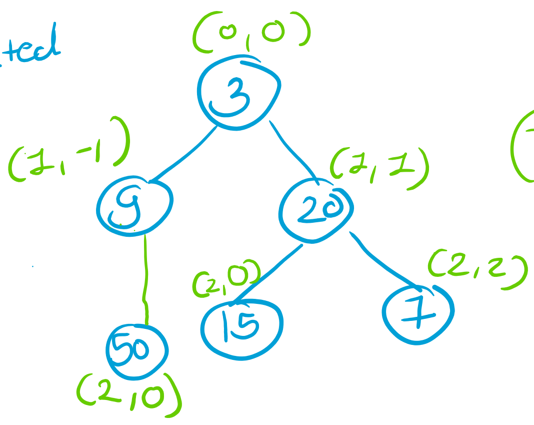


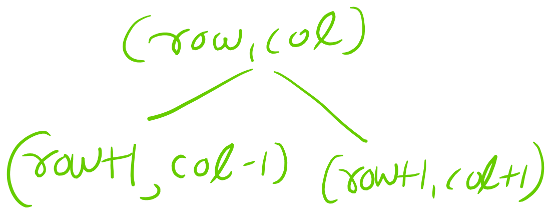
27 March 2024 19:44

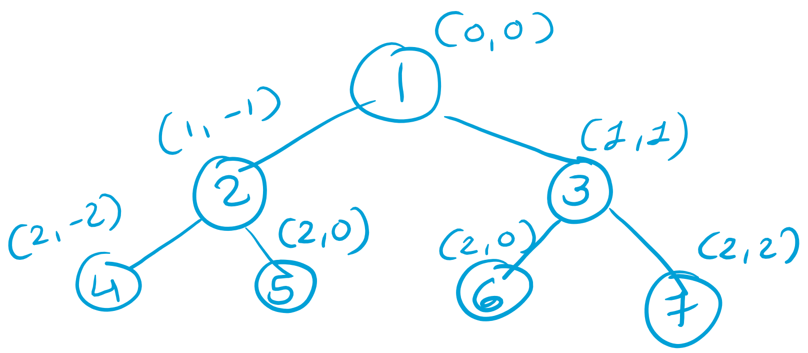
987 vertical Order tra **

if both node codinated are same

-1 = 9 15, 50
0 = 3, 5, 15
1 = 20
2 = 7







-2 = 4
-1 = 2
0 = 1, 5, 6
1 = 3
7 = 2

create a map

map { Col → map<row, multiset> }

sorted

sorted

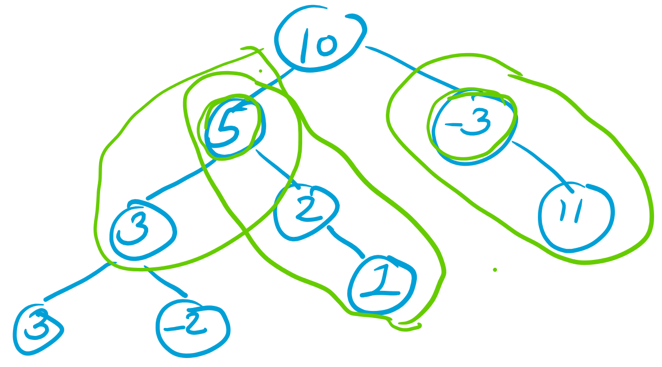
sorted and multiple value are store

Col row val

0 0 → 1

28 March 2024 10:28

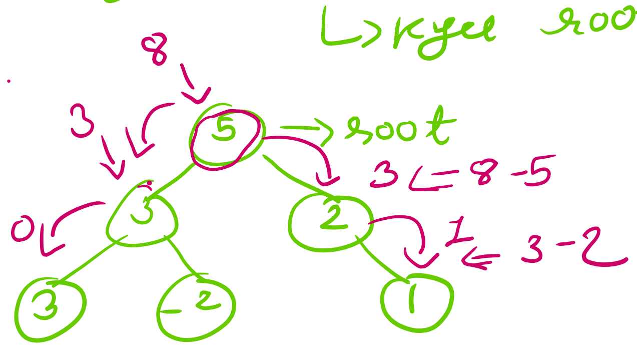
437 K-sum paths



target sum = 8

path = 3

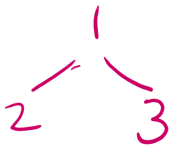
Break it down
↳ kyaa root se path ata hai ?



✱✱
28 March 2024 10:54

94 Morris Tra

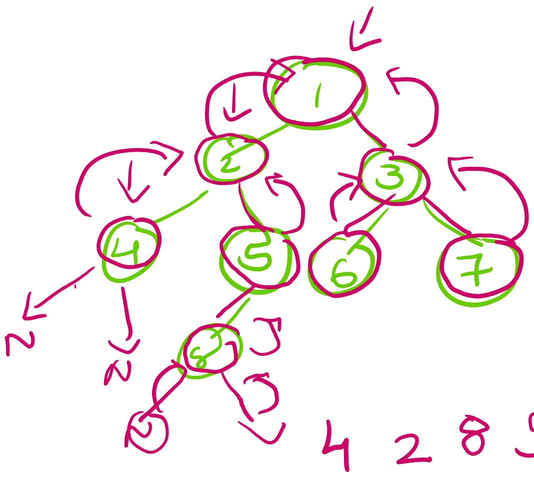
Inorder trav => TC(O(n)) => O(n) } इतना
SC(O(n)) => O(1) }
↑ this is a Morris trd.



2 1 3

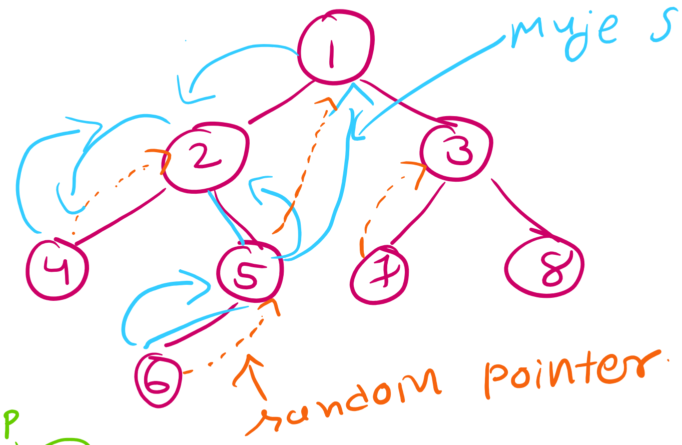
```
inorder(root -> left)
cout << root -> data
inorder(root -> right)
```

Morris Tra => inorder => O(n) = TC
O(1) = SC



inorder

4 2 8 5 1 6 3 7



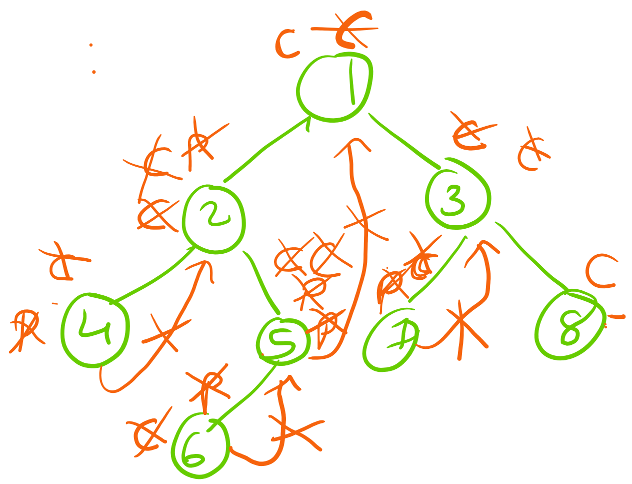
Tree -> left
right
random

random P
4 2 6 5 1 7 3 8
left right

inorder Predecessor
↳ જો કોઈ Node ની પહેલા કઈ Node આવે તે ખબર પડે તો

1 -> 5
7 -> 1

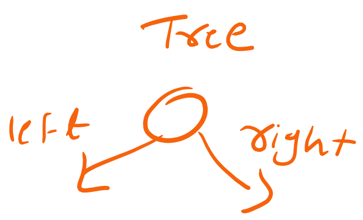
```
while(curr)          curr = root;
{
  // left is null then, visit it
  else {        left ≠ Null
        ① predecessor find
        ② if right Node null then, go left after establisin
                     link from predssor to curr
           else
        ③     left is already visited, go right after
                     visiting curr, & remaning connection
  }
}
```

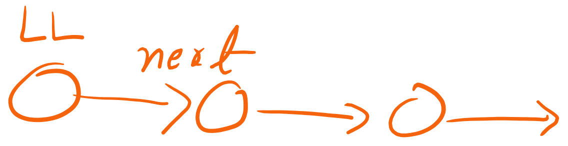


4 2 6 5 1 7 3 8

114 Flutten BT into LL

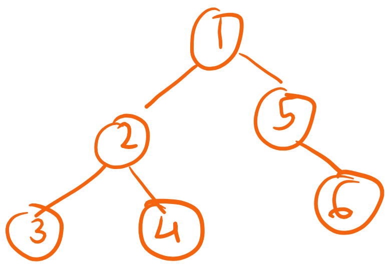
↳ convert

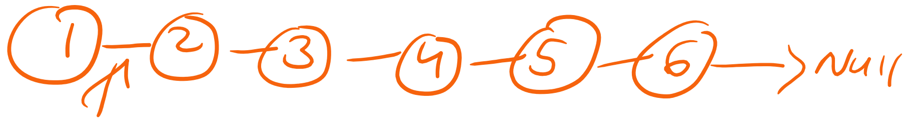




right = next

left = Null



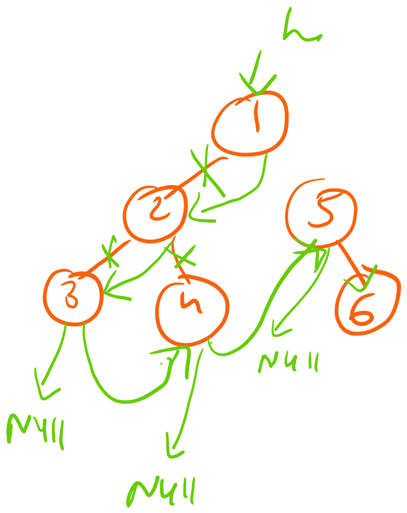


Preorder NLR

1 2 3 4 5 6

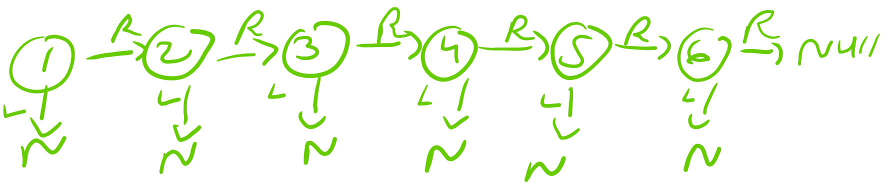
right pointer

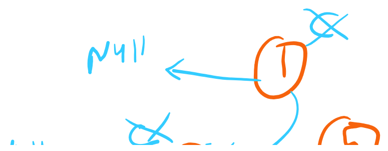
left pointer = NULL



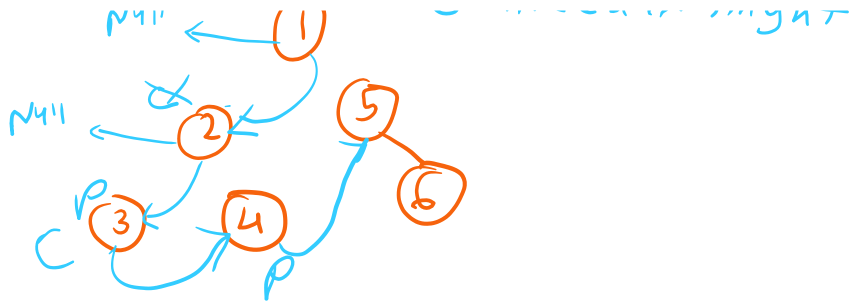
Head ① → root

1 2 3 4 5 6

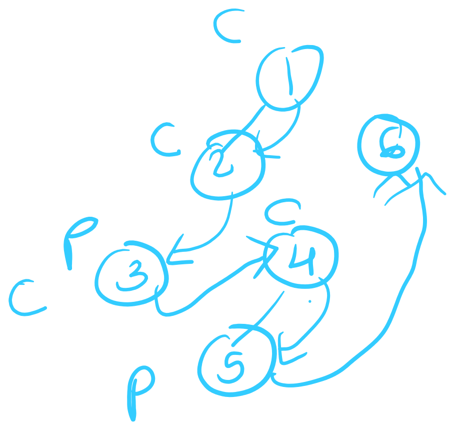




Curr = Curr -> right



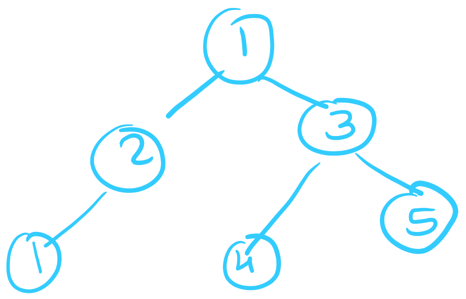
pred->right = curr->right.
curr-> right = curr->left.
curr->left = Null
Move curr = curr->right



1 2 3 4 5 6

28 March 2024 12:03

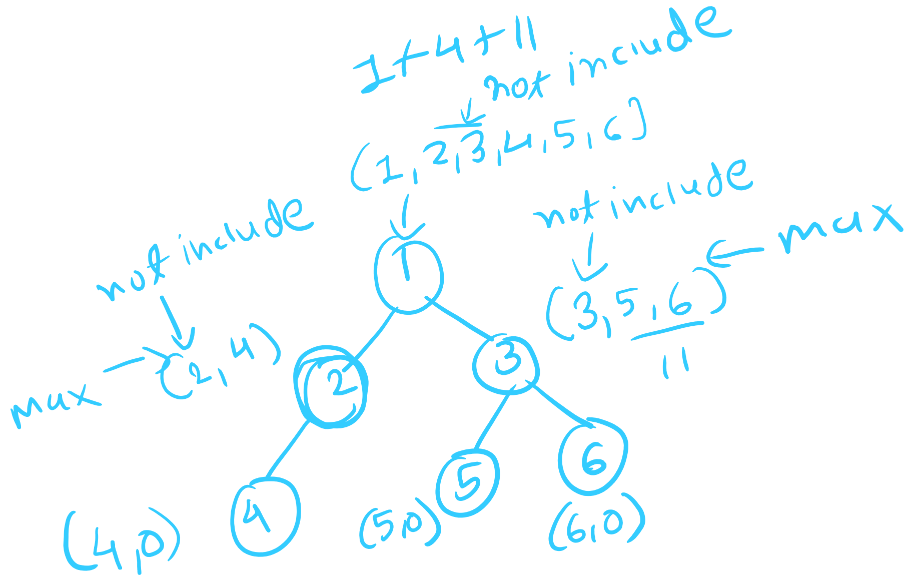
gfg Maximum sum of Non-adjacent Nodes



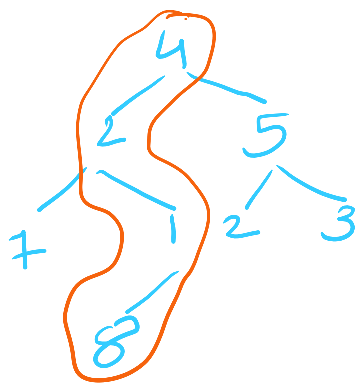
if we choose 1 then that child not choose

C1 = 1, 1, 4, 5 = 11
C2 = 2, 3 = 5
C1 = 1, 4, 5, 1 = 11
C4 = 4, 5, 1, 1 = 11
C5 = 5, 4, 1, 1 = 11

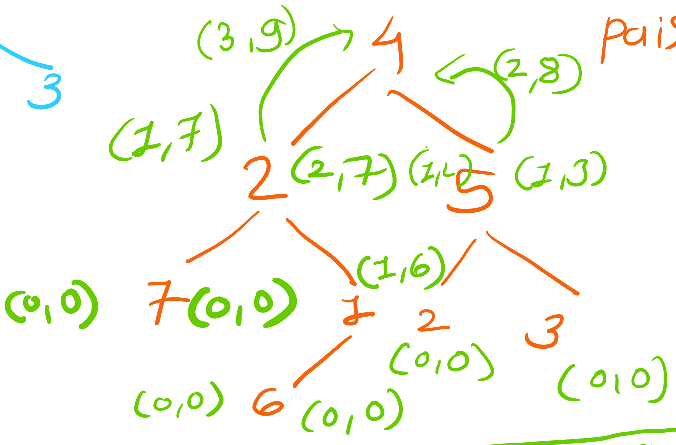
Sum => including all nodes at that level.



gfg Sum of the longest Bloodline of a tree



ans = 4+2+1+8



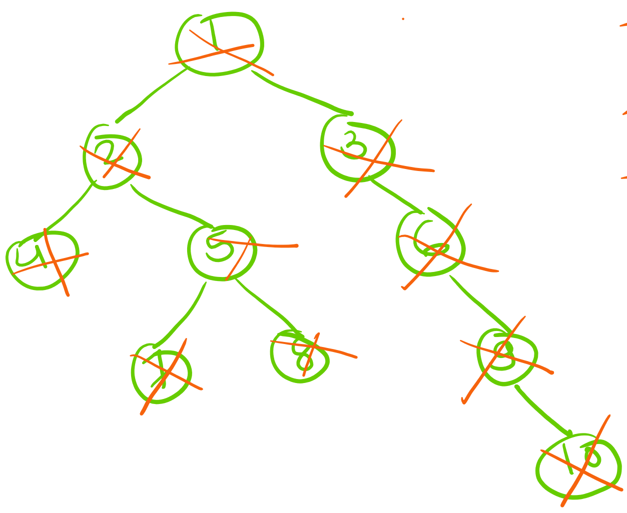
pair => first
↳ height

second
↳ sum

$\boxed{\text{ans} = 9}$

28 March 2024 18:33

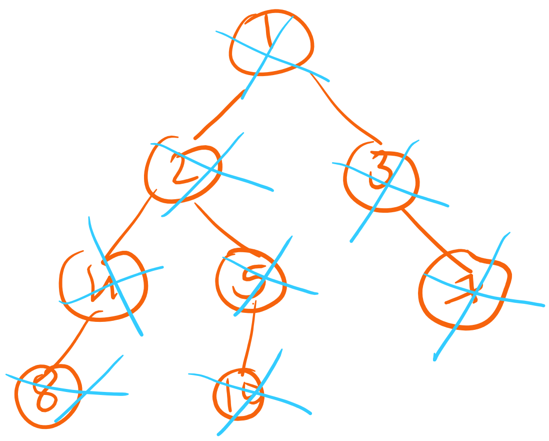
gfg Burning Tree



	burn
t=0,	8
t=1	5
t=2	7,2
t=3	4,1
t=4	3
t=5	6
t=6	9
t=7	10

7 sec burn

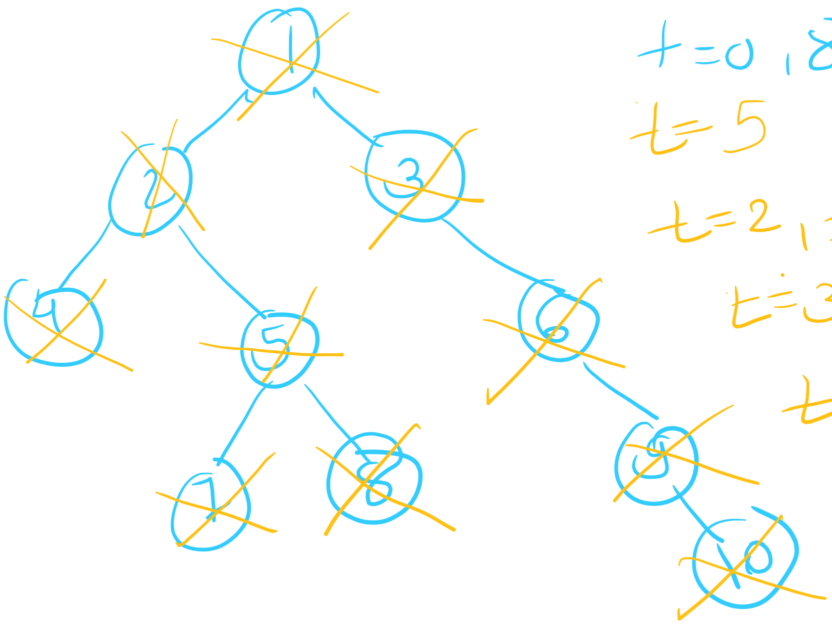
target=10



t=0	, 10
t=1	, 5
t=2	, 2
t=3	, 4, 1
t=4	8, 3
t=5	7

ans t=5

① Find target Node
② make node to parent node mapping



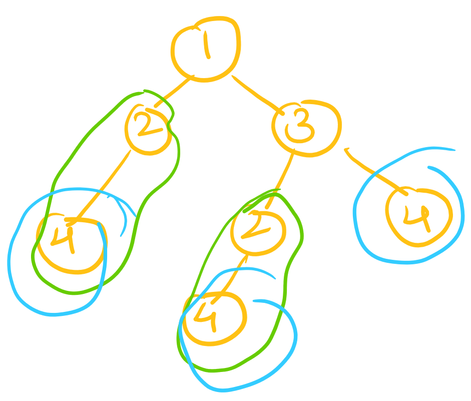
t=0 ,8
t=5
t=2 ,7
t=3
t=4
t=5
t=6
t=7

Visited

queue <= ~~8~~ ~~5~~ ~~2~~, ~~7~~ ~~1~~,~~4~~ ,~~3~~ ,~~6~~ ,~~9~~ ,~~10~~

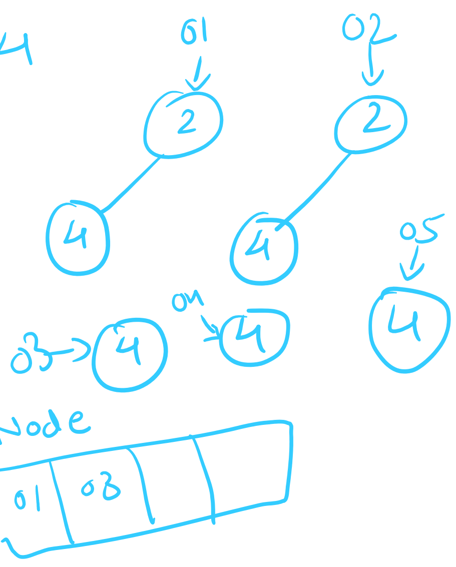
28 March 2024 19:13

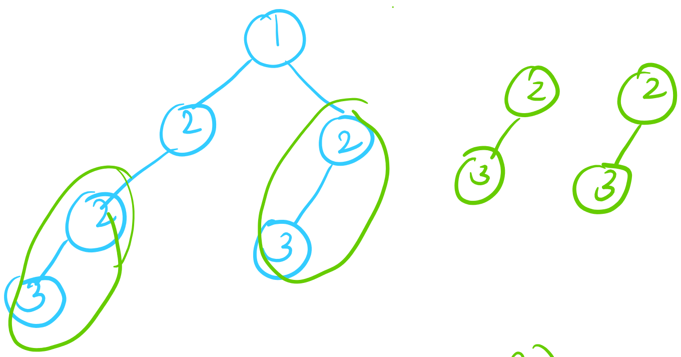
652 Find Duplicate subtrees



2,4

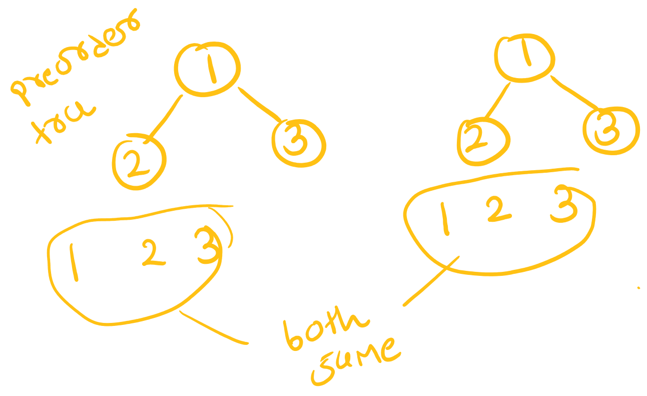
4

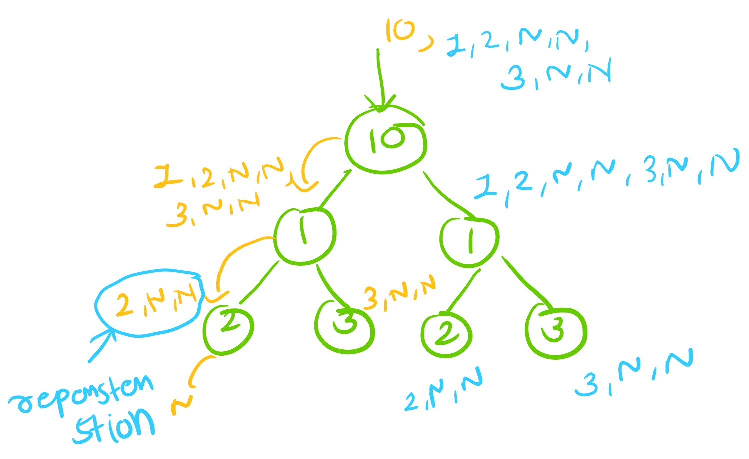




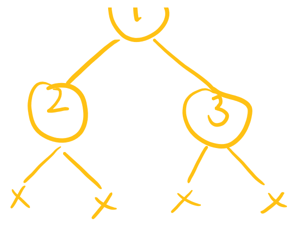
N1 Brute force $O(n^2)$

↳ one node a } then check
other node b

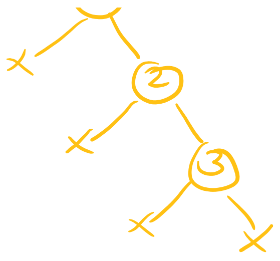




stion



1, 2, N, N, 3, N, N



1, N, 2, N, 3, N, N

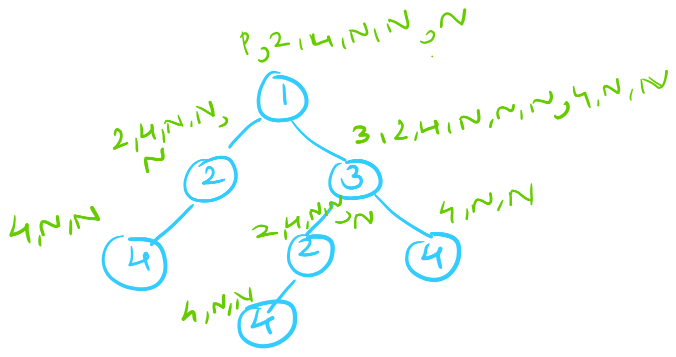
unique
idetify
↓

Map<string, int>

2, N, N	~~1~~ 2
3, N, N	~~1~~ 2
1, 2, N, N, 3, N, N	~~1~~ 2

if Map માં પહેલાથી string પડેલી હોય તો → ans = 2, 3, 1

$\overline{\text{node}}$



ans = 4, 2

4, N, N	~~1~~ ~~2~~ 3
2, 4, N, N, N, N	~~1~~ 2
3, 2, 4, N, N, N, N, 4, N, N, N	1

if 1 equal હોય તો ર ans માં store.