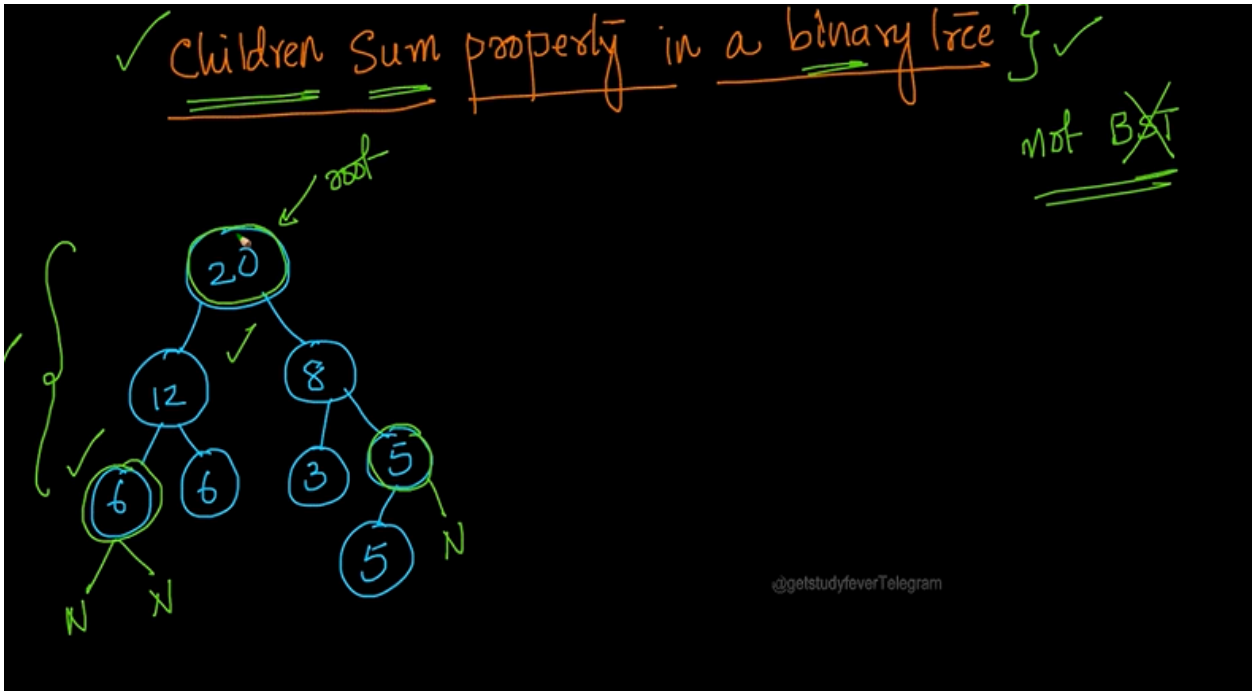
Children Sum Property in binary Tree

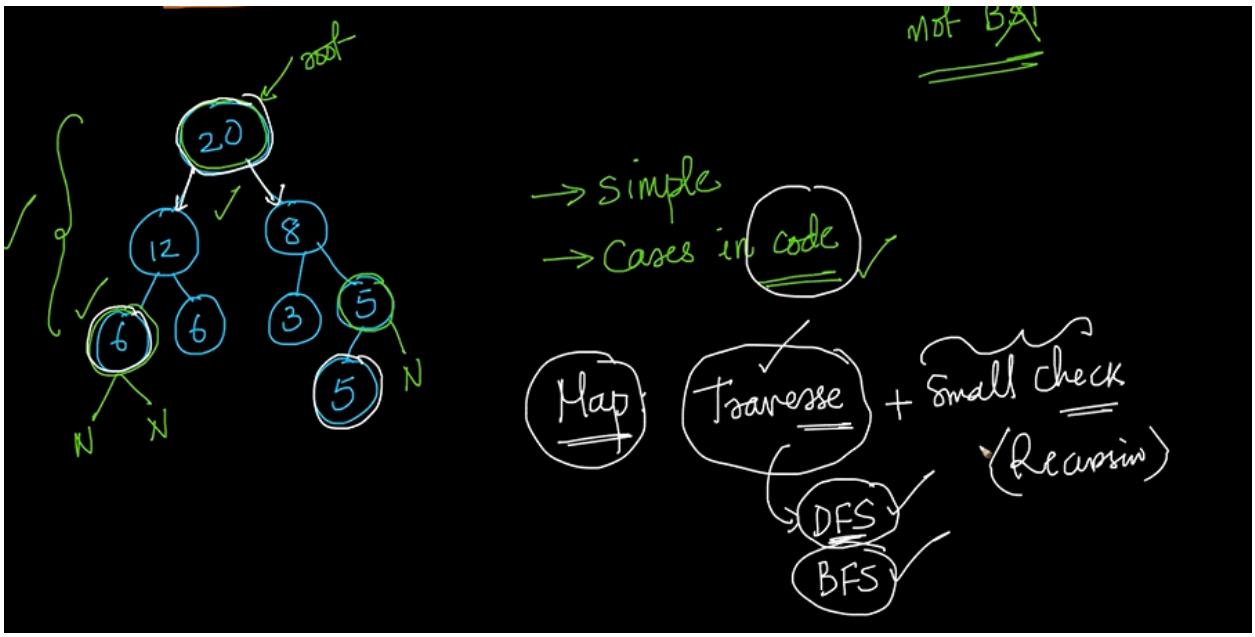
1. Problem statement



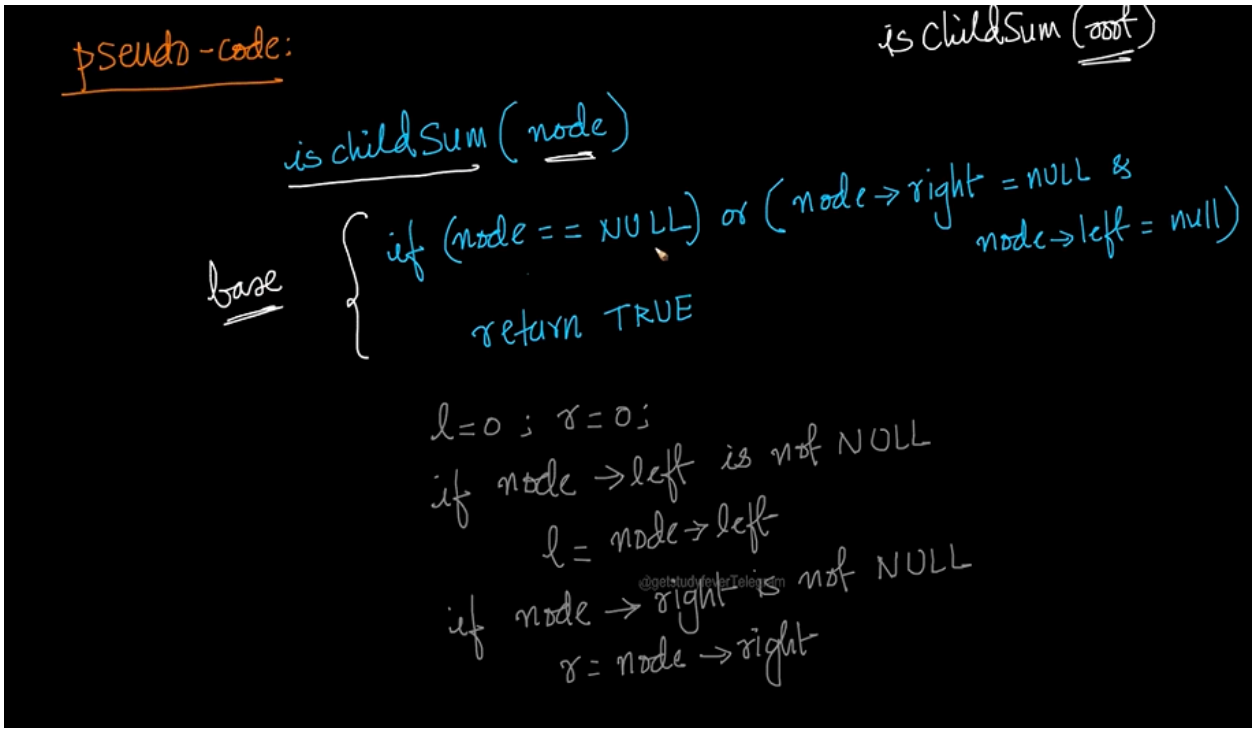
given a binary tree..return True if sum of child nodes equals too parent node

Approach

1. Here we have to handle all the edge cases perfectly
2. We'll go ahead with simple recursion.
3. Here we have to traverse the array and add small check login of sum

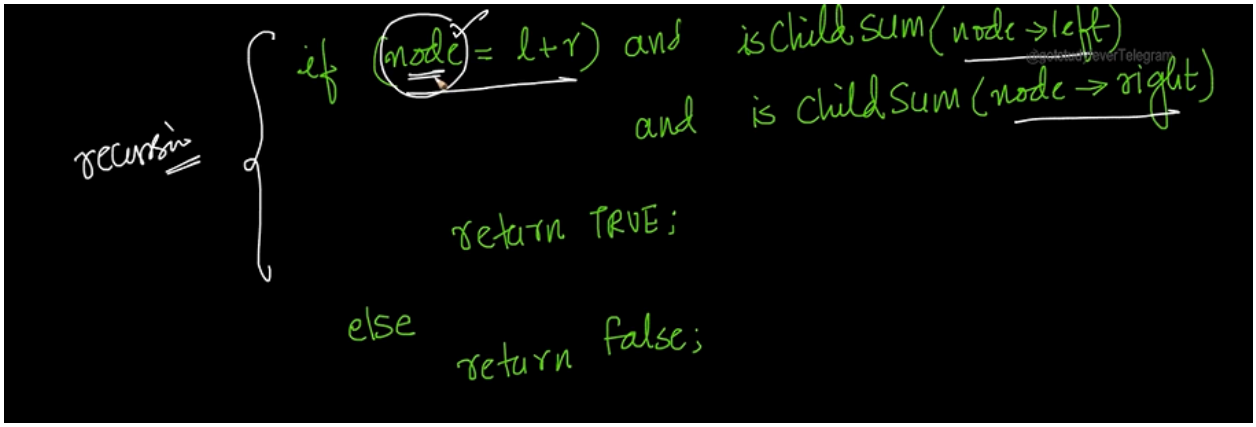


4. And we'll approach this via DFS with recursion
5. Pseudocode



here for any recursion problem ..we have write a base code…see pic

6. And we have used l,r to store the left and right child value…and by default we gave them as l,r = 0,0 ..so incase if there's a null value…then it will be 0



7. Now here given a node…it must satisfy the node = l+r…and its we call our function recursively on its left and right child…if it satisfies..then we return True

Python code:

1.
```
class node:

    def __init__(self,info):
        self.info = info
        self.left = None
        self.right = None


def checkChildSum(root):

    ldata = rdata = 0
    if(root==None or (root.left == None and root.right == None)):
        return True
    else:
        if root.left:
            ldata = root.left.info
        if root.right:
            rdata = root.right.info

        if(root.info == ldata+rdata and checkChildSum(root.left) and checkChildSum(root.right)):
            return True
        else:
            return False

if __name__ == '__main__':
    root = node(15)
    root.left = node(10)
    root.right = node(5)
    root.left.left = node(5)
    root.left.right = node(5)
    root.right.left = node(2)
    root.right.right = node(3)
    if(checkChildSum(root) is True):
        print("children sum property")
    else:
        print("not satisfy children sum property")
```