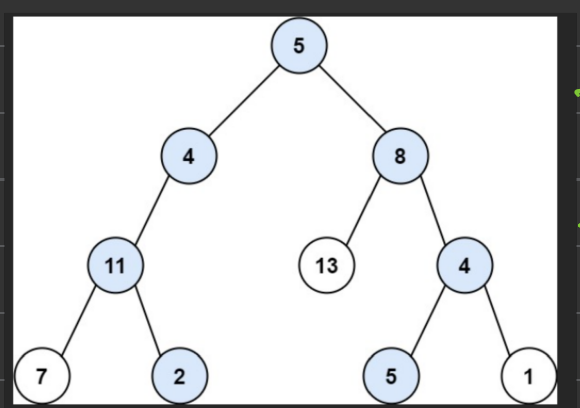
LEETCODE: 113

BY
Lord Priyansh...

Coding Implementation :-

```
class Solution {
public:
    void helper(TreeNode* root,int targetSum,vector<vector<int>> &ans,vector<int> v){
        if (root == NULL)return;
        if(root->left == NULL && root->right == NULL){
            if(targetSum == root->val){
                v.push_back(root->val);
                ans.push_back(v);
            }
            return;
        }
        v.push_back(root->val);
        helper(root->left,targetSum-(root->val),ans,v);
        helper(root->right,targetSum-(root->val),ans,v);
    }
    vector<vector<int>> pathSum(TreeNode* root, int targetSum) {
        vector<vector<int>> ans;
        vector<int> v;
        helper(root,targetSum,ans,v);
        return ans;
    }
};
```

Understanding Problem And thought of Solving...

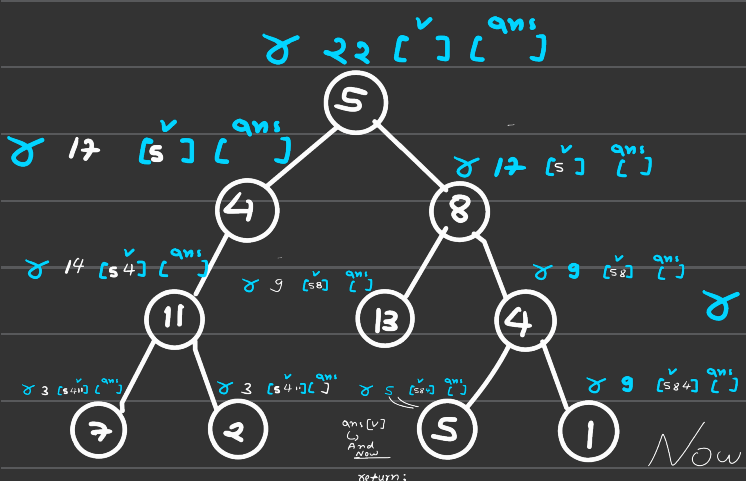


=> we would traverse through the tree nodes using recursion with the help of node named v

=> Before each pass to left or right we would push the node value to vector v and pass it By Value ooo

=> Before calling the left subtree or right subtree using recursion we would subtract the root → val to the previous
↳ current
called value

=> last me if we found k:
root → val == Value passed
↳ signifying ki Aur kitna minus karege to wo zero hoga
↳ example se Better samjhoge...
↳ target sum



ans → By reference

Now finally return ans which have V inside it...