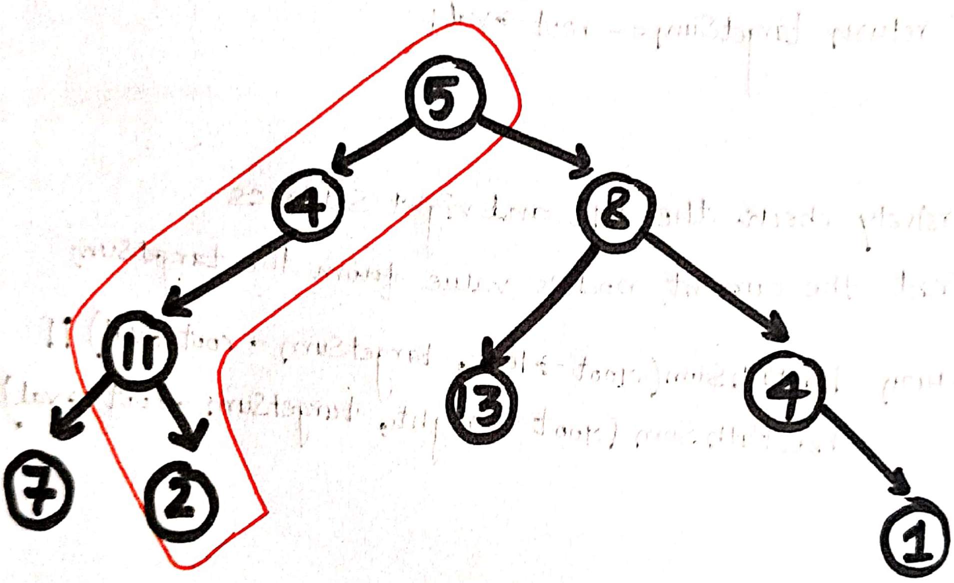
PATH SUM (112)

Given the root of a BINARY TREE AND AN INTEGER targetSum, return true if the tree has a root-to-leaf path such that adding up all the values along the path equals targetSum.



INPUT: root = [5,4,8,11, null, 13,4,7,2, null, null, null, 1],

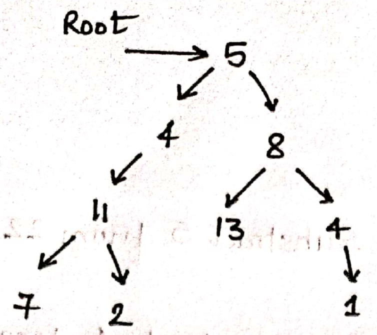
targetSum = 22

OUTPUT: True

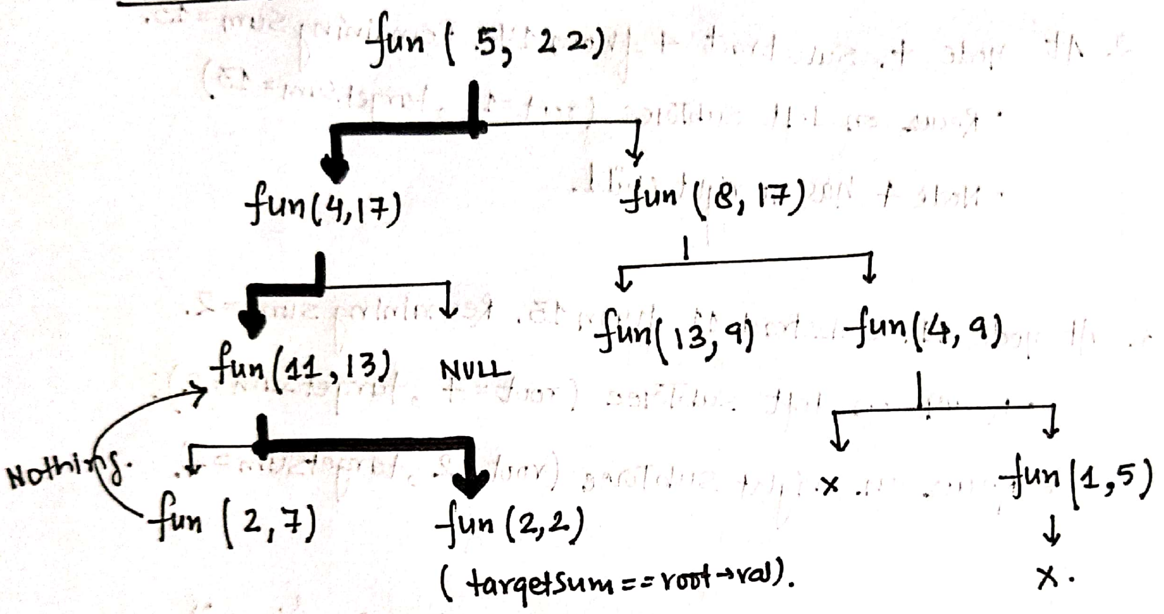
Explaination:→ The root-to-leaf path with the target sum is shown.

C++ Code.

```
class Solution{
public :
    bool hasPathSum (TreeNode* root, int targetSum){
    if(!root) return false;

   // Check if the current node is a leaf node
    if((!root -> left) && (!root -> right)){
   /* // if it's a leaf node, check if the remaining targetSum
     matches its value */

        return targetSum == root->val;

    }

 // Recursively check the left and right SubTrees
 // Subtract the current node's value from the targetSum

      return  hasPathSum(root->left, targetSum - root->val) ||
              has Path Sum (root -> right, targetSum - root ->val);


   }
};
```



Recursion Tree.



5 → 4 → 11 → 2 . Answer.

Target Sum = 22

1. Start at the root (5). Substract 5 from 22.
   Remaining Sum = 17.
   - Recur on leftSubTree (root = 4, targetSum = 17).
   - Recur on right SubTree (root = 8, targetSum = 17).

2. At node 4. Substract 4 from 17. Remaining Sum = 13.
   - Recus on left SubTree (root = 11, targetSum = 13)
   - Node 4 has no right child.

3. At node 11. Substract 11 from 13. Remaining sum = 2.
   - Recur on left subTree (root = 7, targetSum = 2).
   - Recus on right subTree (root = 2, targetSum = 2).

4. At node ② (a leaft node). Remaining sum = 2, Since it matches the node's value, return true.