[**Binary Tree Paths**](https://leetcode.com/problems/binary-tree-paths/)

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** BinaryTreePath {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

TreeNode root = **new** TreeNode(10);

root.left = **new** TreeNode(5);

root.left.left = **new** TreeNode(3);

root.left.right = **new** TreeNode(7);

root.right = **new** TreeNode(15);

root.right.right = **new** TreeNode(18);

System.***out***.println(*binaryTreePaths*(root));

}

**public** **static** List<String> binaryTreePaths(TreeNode root) {

List<String> result = **new** ArrayList<>();

**if**(root == **null**) {

**return** result;

}

*helper*(root , result , "");

**return** result;

}

**public** **static** **void** helper(TreeNode node, List<String> result, String current) {

**if**(node == **null**) {

**return**;

}

**if**(node.left == **null** && node.right == **null**) {

result.add(current + node.val);

**return**;

}

*helper*(node.left , result, current + node.val + "->");

*helper*(node.right , result, current + node.val + "->");

}

}

**class** TreeNode {

**int** val;

TreeNode left;

TreeNode right;

TreeNode(**int** x) { val = x; }

}

Time Complexity : O(n) , n is no of nodes In tree

Space Complexity : O(n), n is no of nodes In tree