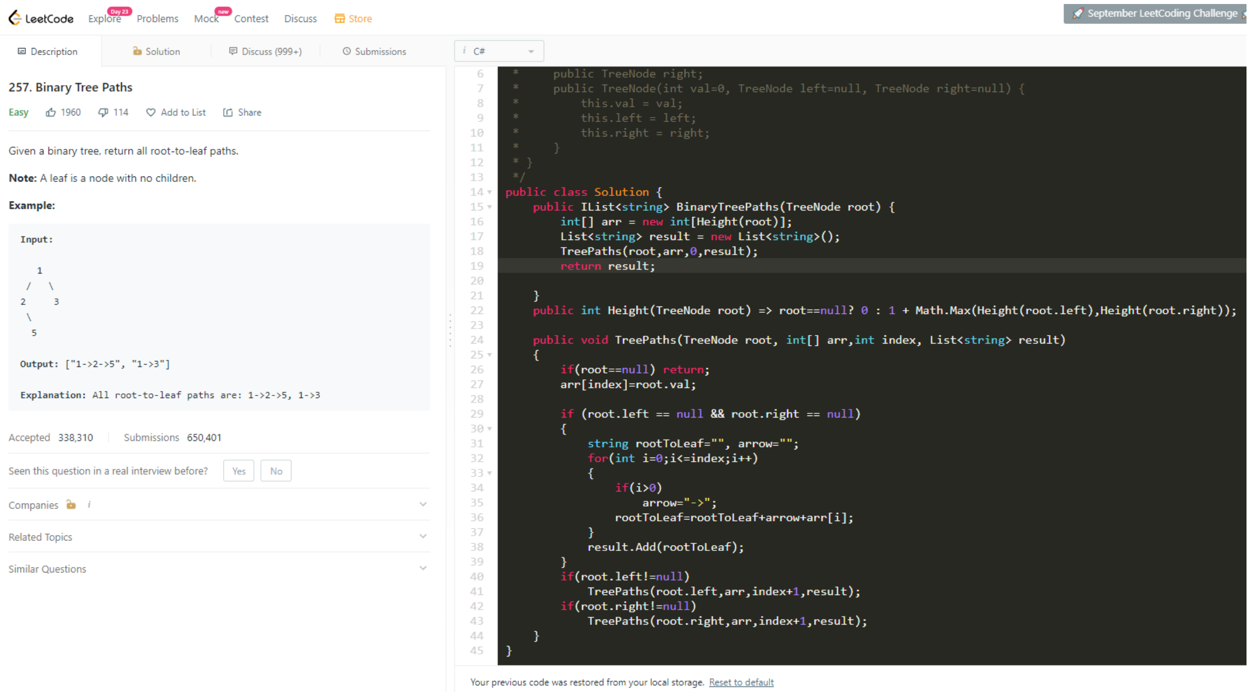
// https://leetcode.com/problems/binary-tree-paths/



257. Binary Tree Paths

Given a binary tree, return all root-to-leaf paths.

Note: A leaf is a node with no children.

Example:

```
Input:

   1
 /   \
2     3
 \
  5

Output: ["1->2->5", "1->3"]

Explanation: All root-to-leaf paths are: 1->2->5, 1->3
```

Accepted 338,310 | Submissions 650,401

Seen this question in a real interview before?

```
 *     public TreeNode right;
 *     public TreeNode(int val=0, TreeNode left=null, TreeNode right=null) {
 *         this.val = val;
 *         this.left = left;
 *         this.right = right;
 *     }
 * }
 */
public class Solution {
    public IList<string> BinaryTreePaths(TreeNode root) {
        int[] arr = new int[Height(root)];
        List<string> result = new List<string>();
        TreePaths(root,arr,0,result);
        return result;

    }
    public int Height(TreeNode root) => root==null? 0 : 1 + Math.Max(Height(root.left),Height(root.right));

    public void TreePaths(TreeNode root, int[] arr,int index, List<string> result)
    {
        if(root==null) return;
        arr[index]=root.val;

        if (root.left == null && root.right == null)
        {
            string rootToLeaf="", arrow="";
            for(int i=0;i<=index;i++)
            {
                if(i>0)
                    arrow="->";
                rootToLeaf=rootToLeaf+arrow+arr[i];
            }
            result.Add(rootToLeaf);
        }
        if(root.left!=null)
            TreePaths(root.left,arr,index+1,result);
        if(root.right!=null)
            TreePaths(root.right,arr,index+1,result);
    }
}
```

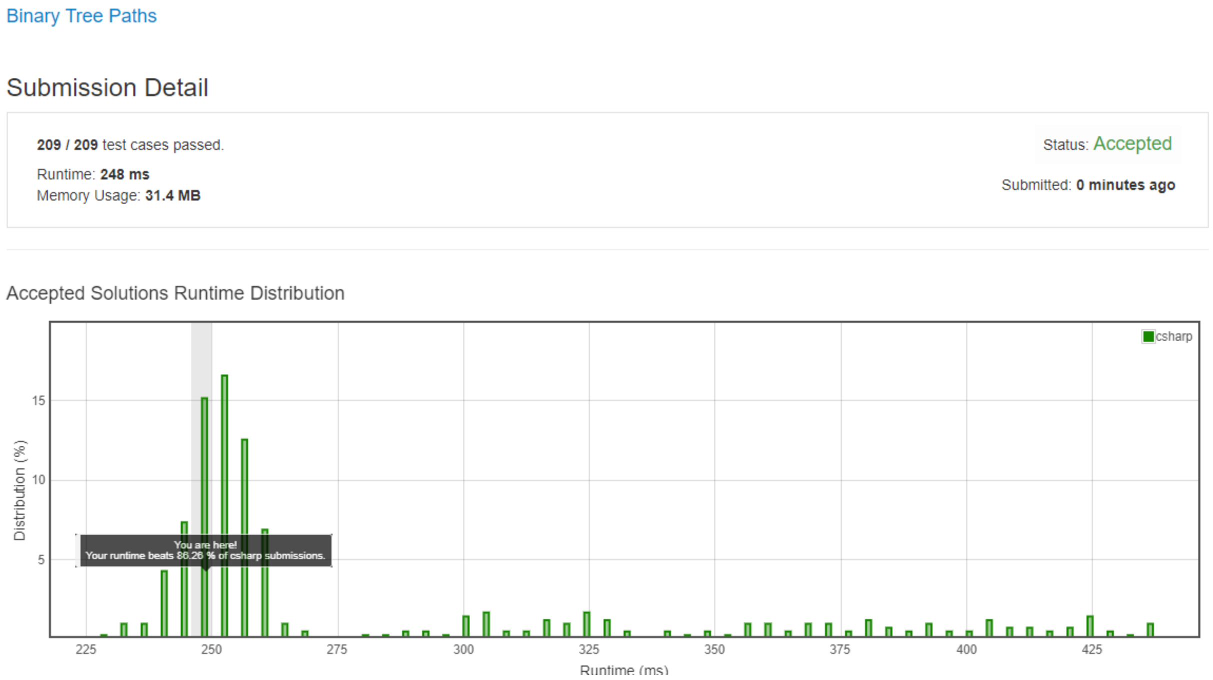
Your previous code was restored from your local storage. Reset to default

```
public class Solution {
    public IList<string> BinaryTreePaths(TreeNode root) {
        int[] arr = new int[Height(root)];
        List<string> result = new List<string>();
        TreePaths(root,arr,0,result);
        return result;

    }
    public int Height(TreeNode root) => root==null? 0 : 1 + Math.Max(Height(root.left),Height(root.right));

    public void TreePaths(TreeNode root, int[] arr,int index, List<string> result)
    {
        if(root==null) return;
        arr[index]=root.val;

        if (root.left == null && root.right == null)
        {
            string rootToLeaf="", arrow="";
            for(int i=0;i<=index;i++)
            {
                if(i>0)
                    arrow="->";
                rootToLeaf=rootToLeaf+arrow+arr[i];
            }
            result.Add(rootToLeaf);
        }
        if(root.left!=null)
            TreePaths(root.left,arr,index+1,result);
        if(root.right!=null)
            TreePaths(root.right,arr,index+1,result);
    }
}
```



Binary Tree Paths

Submission Detail

209 / **209** test cases passed.

Runtime: **248 ms**

Memory Usage: **31.4 MB**

Status: Accepted

Submitted: **0 minutes ago**

Accepted Solutions Runtime Distribution

You are here!
Your runtime beats 86.26 % of csharp submissions.