

# [每日 LeetCode] 257. Binary Tree Paths

作者: [Hanseltu](#)

原文链接: <https://ld246.com/article/1559316171403>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

原文链接 [\[每日LeetCode\] 257. Binary Tree Paths](#)

### Description:

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

**Note:** A leaf is a node with no children.

### Example:

Input:



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

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

---

思路：题目要求输出二叉树的所有路径，考虑递归，以实例为例，主要的递归思路如下：

- 第一次递归：输入 `"`，输出 `'1'`。
- 第二次递归：输入 `'1'`，左子结点输出 `'1->2'`，右子结点输出 `'1->3'`（右子结点为叶子结点，停止递归）。
- 第三次递归：输入 `'1->2'`，左子结点的右子结点输出 `'1->2->5'`（左子结点的右子结点为叶子结点，停止递归）。
- 输出：`'1->3'`和`'1->2->5'`。

---

### C++代码

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
class Solution {
public:
    void binaryTreePaths(vector<string>& result, TreeNode* root, string t){
        if(!root->left&&!root->right){
            result.push_back(t);
            return;
        }
    }
};
```

```
    if(root->left)
        binaryTreePaths(result,root->left,t+"->" +to_string(root->left->val));
    if(root->right)
        binaryTreePaths(result,root->right,t+"->" +to_string(root->right->val));
}
vector<string> binaryTreePaths(TreeNode* root) {
    vector<string> vec;
    if(!root) return vec;
    binaryTreePaths(vec,root,to_string(root->val));
    return vec;
}
};
```

---

运行时间: 8ms

运行内存: 11.7M