

[LeetCode] Binary Tree Paths - 二叉树基础系列题目

原创

Eastmount

2015-09-09 00:58:21

3294

☆ 收藏

版权

分类专栏:

LeetCode

文章标签:

leetcode

二叉树



Python+TensorFlow人工智能

该专栏为人工智能入门专栏，采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法、神...



Eastmount

¥9.90

订阅博主

目录:

- 1.Binary Tree Paths - 求二叉树路径
- 2.Same Tree - 判断二叉树相等
- 3.Symmetric Tree - 判断二叉树对称镜像

Binary Tree Paths

题目概述:

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

For example, given the following binary tree:

```
  1
 /  \
2    3
 \
  5
```

All root-to-leaf paths are:

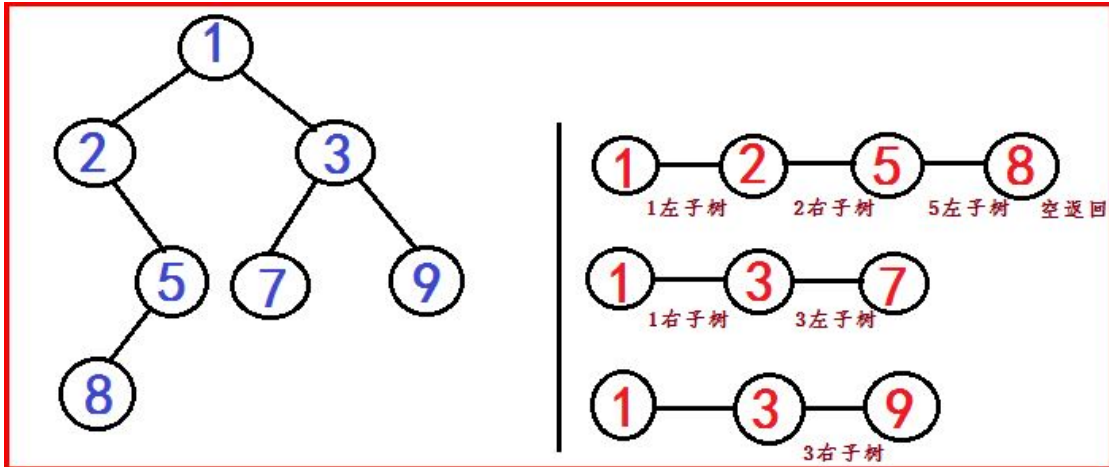
```
["1->2->5", "1->3"]
```

题目解析:

本题主要考察二叉树遍历操作，输出二叉树的所有路径，通常采用递归方法能很好的解决。但是如果采用C语言编写，返回二维字符串数组如何添加二叉树路径是个难点？

```
char** binaryTreePaths(struct TreeNode* root, int* returnSize) {}
```

最终采用C++完成，当遍历至叶子节点时，通过容器push_back添加一条路径。



我的代码：

```
/**
 * 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:
    //创建空容器 对象类型为string类
    vector<string> result;
    void getPaths(TreeNode* node, string path) {
        if(node->left==NULL && node->right==NULL) { //左右子树为空 路径寻找完成 增加
至数组中
            result.push_back(path);
        }
        if(node->left!=NULL) { //递归遍历左子树 当前路径添加左孩子结点
            getPaths(node->left, path+"->"+to_string(node->left->val));
        }
        if(node->right!=NULL) { //递归遍历右子树
            getPaths(node->right, path+"->"+to_string(node->right->val));
        }
    }
}

//获取二叉树路径
vector<string> binaryTreePaths(TreeNode* root) {
    if(root==NULL)
        return result;
}
```