Problem Link:

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

Solution:

/\*\*

\* Definition for a binary tree node.

\* struct TreeNode {

\* int val;

\* TreeNode \*left;

\* TreeNode \*right;

\* TreeNode() : val(0), left(nullptr), right(nullptr) {}

\* TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}

\* TreeNode(int x, TreeNode \*left, TreeNode \*right) : val(x), left(left), right(right) {}

\* };

\*/

class Solution {

public:

vector<string> result;

void dfs(TreeNode\* node, string path)

{

if(!node)

return;

if(!path.empty())

path += "->";

path += to\_string(node->val);

if(!node->left && !node->right)

{

result.push\_back(path);

return;

}

dfs(node->left, path);

dfs(node->right, path);

}

vector<string> binaryTreePaths(TreeNode\* root)

{

dfs(root, "");

return result;

}

};