# 1372. Longest ZigZag Path in a Binary Tree

You are given the root of a binary tree.

A ZigZag path for a binary tree is defined as follow:

Choose any node in the binary tree and a direction (right or left).

If the current direction is right, move to the right child of the current node; otherwise, move to the left child.

Change the direction from right to left or from left to right.

Repeat the second and third steps until you can't move in the tree.

Zigzag length is defined as the number of nodes visited - 1. (A single node has a length of 0).

Return the longest ZigZag path contained in that tree.

## SOLUTION IN C++

struct T {

int leftMax;

int rightMax;

int subtreeMax;

};

class Solution {

public:

int longestZigZag(TreeNode\* root) {

return dfs(root).subtreeMax;

}

private:

T dfs(TreeNode\* root) {

if (root == nullptr)

return {-1, -1, -1};

const T left = dfs(root->left);

const T right = dfs(root->right);

const int leftZigZag = left.rightMax + 1;

const int rightZigZag = right.leftMax + 1;

const int subtreeMax =

max({leftZigZag, rightZigZag, left.subtreeMax, right.subtreeMax});

return {leftZigZag, rightZigZag, subtreeMax};

}

};