**SOLUTION**

/\*\*

\* 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:

int max\_a=-1;

int diameter(TreeNode\* root){

if(!root) return 0;

if(!root->left && !root->right) return 0;

int left\_m=0;

if(root->left)

left\_m=1+diameter(root->left);

int right\_m=0;

if(root->right)

right\_m=1+diameter(root->right);

if(max\_a<left\_m+right\_m)

max\_a=left\_m+right\_m;

return max(left\_m,right\_m);

}

int diameterOfBinaryTree(TreeNode\* root) {

return max(max\_a,diameter(root));

}

};

**TIME COMPLEXITY: O(N)**

**SPACE COMPLEXITY: O(1)**