103. Binary Tree Zigzag Level Order Traversal

Medium

Given a binary tree, return the *zigzag level order* traversal of its nodes' values. (ie, from left to right, then right to left for the next level and alternate between).

For example:

Given binary tree [3,9,20,null,null,15,7],

3  
 / \  
 9 20  
 / \  
 15 7

return its zigzag level order traversal as:

[  
 [3],  
 [20,9],  
 [15,7]  
]

/\*\*

\* 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:

vector<vector<int>> zigzagLevelOrder(TreeNode\* root) {

vector<vector<int>> vec;

if(root==NULL) return vec;

int childCount=1;

queue<TreeNode\*> q;

q.push(root);

bool reverse=false;

while(!q.empty()){

int c=0;

vector<int> curr\_vec;

for(int i=0;i<childCount;i++){

TreeNode\* curr=q.front();

q.pop();

if(!reverse) curr\_vec.push\_back(curr->val);

else curr\_vec.insert(curr\_vec.begin(),curr->val);

if(curr->left!=NULL){

q.push(curr->left);

c++;

}

if(curr->right!=NULL){

q.push(curr->right);

c++;

}

}

vec.push\_back(curr\_vec);

childCount=c;

reverse=(reverse==false)?true:false;

}

return vec;

}

};

Success

[Details](https://leetcode.com/submissions/detail/210654490/)

Runtime: 8 ms, faster than 100.00% of C++ online submissions for Binary Tree Zigzag Level Order Traversal.

Memory Usage: 13.5 MB, less than 71.14% of C++ online submissions for Binary Tree Zigzag Level Order Traversal.