

09/05/24

Lab 2:

Zig-Zag Traversal of BST

```
int ** zigzagLevelOrder (struct TreeNode * root,  
int * returnSize, int ** returnColumnSize) {  
    int th = funheight (root);  
    int ** ret = (int **) calloc (th, sizeof (int **));
```

```
    *returnSize = th;  
    (*returnColumnSize) = (int *) calloc (th, sizeof  
    (int));
```

```
    for (int i = 0; i < th; i++) {  
        int cnt = 0;  
        ret[i] = (int *) calloc (1 < i, sizeof (int));  
        if (i % 2 < 0)  
            funR (root, i, ret[i], &cnt);  
        else  
            funL (root, i, ret[i], &cnt);  
        (*returnColumnSize)[i] = cnt;  
    }
```

```
void funR (struct TreeNode * root, int level,  
int * ret, int * cnt) {  
    if (!root) return;  
    if (level == 0) {  
        ret[( * cnt)++] = root->val;  
    } else {  
        funR (root->right, level - 1, ret, cnt);  
        funR (root->left, level - 1, ret, cnt);  
    }  
    return;  
}
```

```

void funL(struct Treenode* root,
    int level, int *ret, int *cnt) {
    if(!root) return;
    if(level == 0) {
        ret[*cnt++] = root->val;
    }
    else {
        funL(root->left, level-1, ret, cnt);
        funR(root->right, level-1, ret, cnt);
    }
    return;
}

```

Output:

Case 1:

root = [3, 9, 20, null, null, 15, 7]

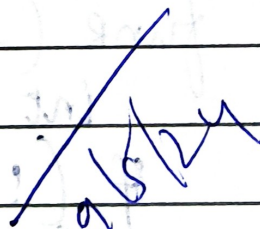
Output:

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

Case 2:

root = [1]

Output: [[1]]



Case 3:

root = []

Output: []