

Leetcode -> 102.

Binary Tree ZigZag Level Order

```

int** zigzagLevelOrder(struct TreeNode* root, int*
    returnSize, int** returnColumnSize){
    int **ans = malloc(2000 * sizeof(int));
    *returnColumnSize = malloc(2000 * sizeof(int));
    *returnSize = 0;
    struct TreeNode *tmp[2000] = {0};
    int top = -1, start = 0;
    tmp[++top] = root;
    while (tmp[start]) {
        int tmp_top = top;
        ans[(*returnSize)] = malloc((top - start + 1) *
            sizeof(int));
        (*returnColumnSize)[(*returnSize)] = (top - start + 1);
        int idx = (*returnSize) % 2 ? (top - start + 1) - 1 : 0;
        int step = (*returnSize) % 2 ? -1 : 1;
        while (start <= tmp_top) {
            ans[(*returnSize)][idx] = tmp[start] -> val;
            if (tmp[start] -> left)
                tmp[++top] = tmp[start] -> left;
            if (tmp[start] -> right)
                tmp[++top] = tmp[start] -> right;
            start++;
            idx += step;
        }
        (*returnSize)++;
    }
    return ans;
}

```

Case 1 ⇒

Input.

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

Output

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

Expected

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

Case 2 ⇒

Input

root = [1]

Output

[[1]]

Expected

[[1]]

Case 3 ⇒

Input

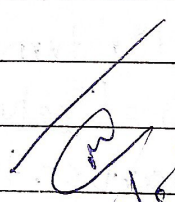
root = []

Output

[]

Expected

[]


9/5/24