

29/5/24

Leet Code 2

Binary Tree Zigzag Level Order Traversal

Code:-

```
int ** zigzagLevelOrder (struct TreeNode *
                        root, int * returnSize, int **
                        returnColumnSizes)
{
    int ** ans = malloc (2000 * sizeof (int *));
    *returnColumnSizes = malloc (2000 * sizeof (int));
    *returnSize = 0;
    struct TreeNode * temp [2000] = {0};
    int top = -1, start = 0;
    temp [++top] = root;

    while (temp [start])
    {
        int temp_top = top;
        ans [( * returnSize)] = malloc (top - start + 1)
        * sizeof (int));
        (*returnColumnSizes) [( * returnSize)] =
        (top - start + 1);

        int idx = ( * returnSize) % 2 ? (top - start + 1)
        - 1 : 1;

        while (start <= temp_top)
        {
            ans [( * returnSize)] [idx] = temp [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;
}

```

Test cases

Case 1 :-

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

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

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

Case 2 :-

Input : [1]

Output : [[1]]

Expected : [[1]]

Case 3

Input = []

Output = []

Expected = []

9/15/24