Algorithm Print the **Zig Zag** Traversal of a Binary Tree

Ensure: The tree is **not** empty

```
1: function ZIG_ZAG_TRAVERSAL(root)
   ▷ current_stack contains all the elements of the current level.
   ▷ next_stack contains all the elements of the next level.
   ▷ left_to_right is true if the current level associates from left to right

2:     current_stack.push(root)
3:     left_to_right ← True
4:     while current_stack is not empty, do
5:         while current_stack is not empty, do
6:             node ← current_stack.top
7:             current_stack.pop
8:             Print(node.data)
9:             if left_to_right then
10:                if Left Child Exists then
11:                    next_stack.push(node.left)
12:                if Right Child Exists then
13:                    next_stack.push(node.right)
14:            else
15:                if Right Child Exists then
16:                    next_stack.push(node.right)
17:                if Left Child Exists then
18:                    next_stack.push(node.left)
19:        Print(One level has been printed)
20:        swap(current_stack, next_stack)                ▷ Go to the next level
21:        left_to_right ←!left_to_right                  ▷ Change the associativity of the level
```