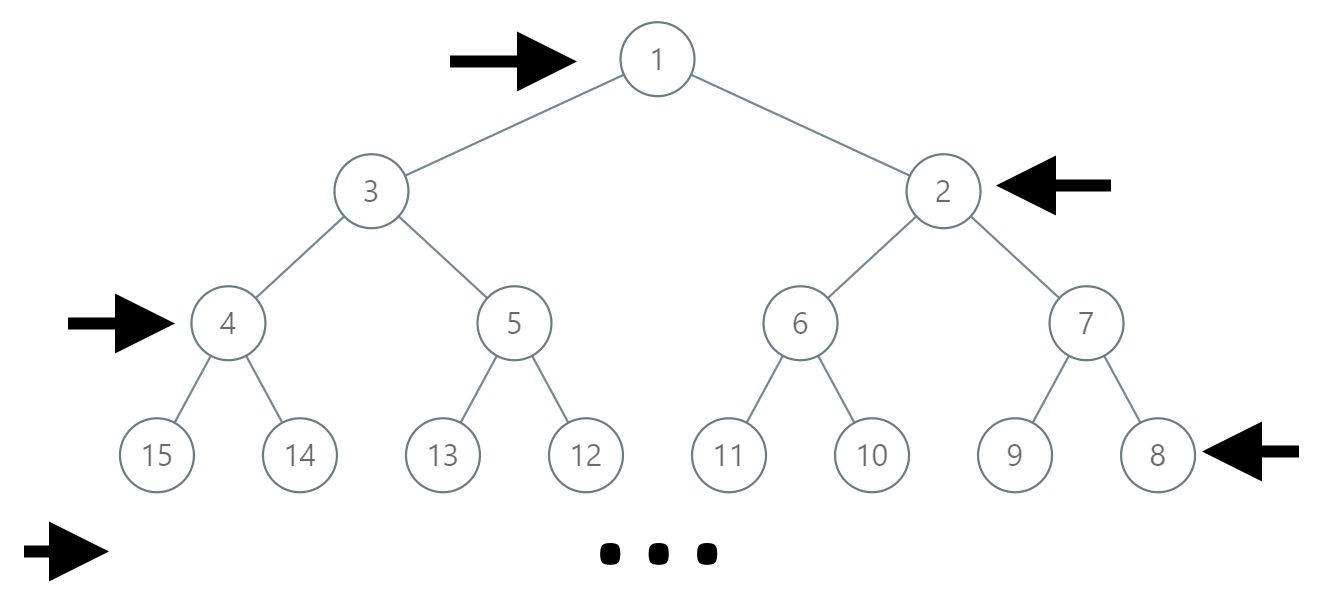
<https://leetcode.com/problems/path-in-zigzag-labelled-binary-tree/>

given an infinite binary tree where every node has two children, the nodes are labelled in row order.

In the odd numbered rows (ie., the first, third, fifth,...), the labelling is left to right, while in the even numbered rows (second, fourth, sixth,...), the labelling is right to left.



Given the label of a node in this tree, return the labels in the path from the root of the tree to the node with that label.

**Example 1:**

**Input:** label = 14

**Output:** [1,3,4,14]

**Example 2:**

**Input:** label = 26

**Output:** [1,2,6,10,26]

Approach:

Brute Force:

As you understand the simplest approach would be just generate the binary tree like above, and do the traversal and find the nodes in between.

Optimal Approach:

Let’s understand the question first,

1. Odd Level : When level is odd, numbering start from right
2. Even Level : When level is Even, numbering start from left

Now for a moment forget about the zig-zag tree, and have a normal tree where numbering is based on node insert in level order traversal

In Full binary tree, if you are at a node (say 6) we can find its parent by 6/2 = 3 [ because each node have 2 child’s, then they must be separated by 2 child’s only at each level].

For 7 -> 7/2 = 3

For 5/2= 2

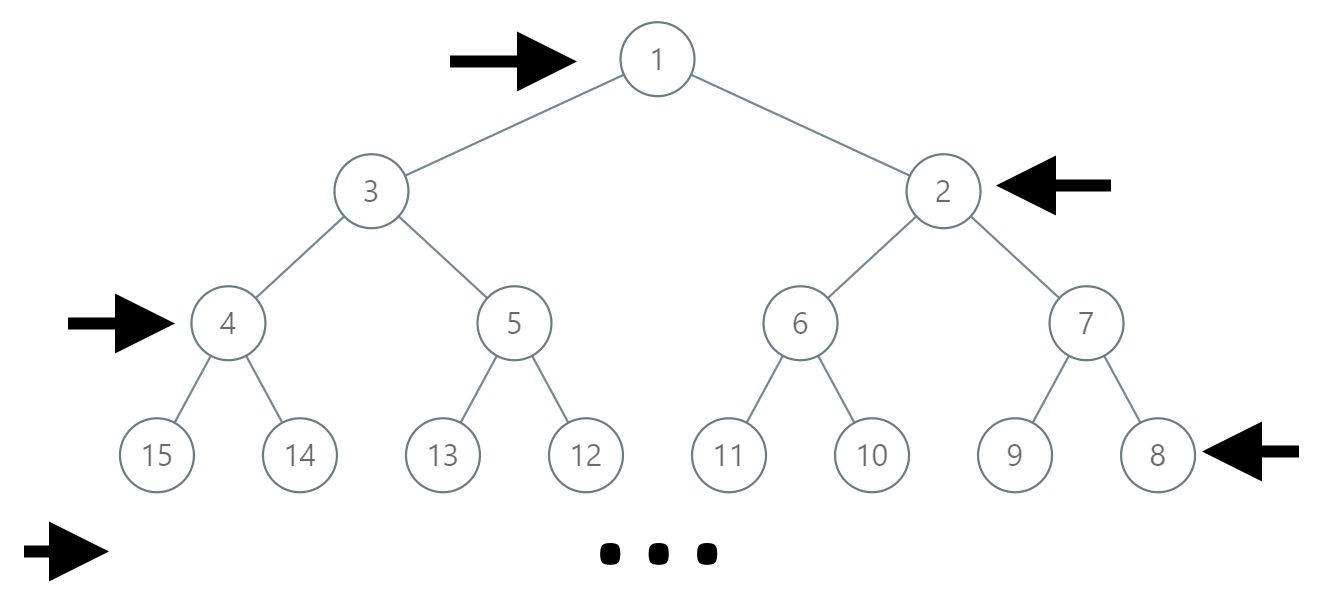
Back to original question.

The only difference with the normal tree vs zig-zag tree has the numbering getting altered based on level (even/odd).

Our task is boils down to first find the level, at which our target is placed in zig-zag tree.

If you observe, then you’ll see its regardless that tree is normal or zig-zag the placement of any label will always be based on Hight of tree [ level of tree ].

So in original example



The level of 14 can be easily evaluated by 2^n >= 14. The least value of n is your level.

Here in our case, n = 4 as 2^4 = 16 > 14 [ whereas 2^3 < 14 ].

Ok, now we know what is the level of our current label = 4.

Next we need to know, what is the level at which its parent would be, and that is trivial it will always be 1 less than its level until you hit root.

But since the tree is zig-zag, just doing 14/2 = 7 wont’ give you the parent.

But it will tell you the where could be the parent.

See at level 3 [ parent level] we have 4 , 5, 6, 7 which tells 7 is symmetry of 4 and 6 is for 5 and vice-versa.

i.e. 14/2 = 7 then our parent would be 4 [ 4 is symmetry for 7 ].

There could be multiple way to find this

1. Using Level with odd-even rule
2. By applying maths in it.

I’m covering first 2nd approach.

Ok, now we know this level have 4,5,6,7 as nodes

And if you do (7 + 4 – 4)/2 = (7)/2 = 7/2=3 [your required parent of 4].

Means**, (level last node value + level first node value – lastLable)/2**

**Ok, we know at each level how many nodes are there which is defined by**

**2^(level-1) ; for level = 3 => 2^2 = 4.[ which is the start of this level in normal tree]**

**For end of the level we can simply do 2^(level) -1 = 8-1 = 7**

**Ok, so we have a formula to calculate the last node label and first node label using current level.**

**( ( 2^level -1 ) + ( 2^(level-1) ) - lable ) /2**

Hence;

Label=14 with level 4 => (2^4 – 1 + 2^3 – 14)/2 = (15 + 8 – 14 )= 9/2 = 4

Label = 4 with level 3 => (7 + 4 – 4 ) / 2 = 3

Label = 3 with level 2 => (4 + 2 -3 )/2 = 3/2 = 1 [your main root]

Hence [1,3,4,14]