RIGHT/LEFT SIDE VIEW OF A BINARY TREE

Problem asks us to display/output the side view (left or right) of a Binary Tree.

Exact opposite of Top/Bottom view problem. Here we just have to show the first element of each level (if left view).

We create a recursive function which traverses the binary tree in a format where you prioritize right child over left. We also maintain an answer vector and push to it everytime we encounter a level for the first time.
This works out because since we are prioritising right child during traversal, the first time we will ever encounter a level will be at its rightmost node.

Pseudocode :

```
recursion(Node * root, int level, vector<int> &res) {
    if(root == NULL) {
        return;
    }
    if(res.size() == level) {
```

```
        res.push_back(root->val);
    }
    recursion(root->right, level+1, res);
    recursion(root->left, level+1, res);
}

rightSideView(Node * root) {
    vector<int> ans;
    recursion(root, 0, ans);
    return ans;
}
```