```java
// Function to return a list of nodes visible from the top view
// from left to right in Binary Tree.
// Hint: Vertical Order Traversal. Pair{Node node, int hd}
static ArrayList<Integer> topView(Node root) {
    Queue<Pair> queue = new LinkedList<>();
    Hashtable<Integer, Node> table = new Hashtable<>();
    queue.add(new Pair(root, 0));
    int min = 99999999;
    int max = -99999999;
    while (!queue.isEmpty()) {
        Pair p = queue.remove();
        if (!table.containsKey(p.hd)) {
            table.put(p.hd, p.node);
        }
        min = Math.min(min, p.hd);
        max = Math.max(max, p.hd);
        if (p.node.left != null) {
            queue.add(new Pair(p.node.left, p.hd - 1));
        }
        if (p.node.right != null) {
            queue.add(new Pair(p.node.right, p.hd + 1));
        }
    }
    ArrayList<Integer> list = new ArrayList<>();
    for (int i = min; i <= max; i++) {
        if (!table.containsKey(i)) {
            continue;
        }
        list.add(table.get(i).data);
    }
    return list;
}
```