```java
// Function to return a list containing the bottom view of the given binary tree.
// Node{int data, int hd, Node left, Node right}
public ArrayList<Integer> bottomView(Node root) {
    Hashtable<Integer, ArrayList<Node>> table = new Hashtable<>();
    Queue<Node> queue = new LinkedList<>();
    root.hd = 0;
    queue.add(root);
    int min = 99999999, max = -99999999;
    while (!queue.isEmpty()) {
        Node u = queue.remove();
        min = Math.min(min, u.hd);
        max = Math.max(max, u.hd);
        if (table.containsKey(u.hd)) {
            table.get(u.hd).add(u);
        } else {
            ArrayList<Node> list = new ArrayList<>();
            list.add(u);
            table.put(u.hd, list);
        }
        if (u.left != null) {
            u.left.hd = u.hd - 1;
            queue.add(u.left);
        }
        if (u.right != null) {
            u.right.hd = u.hd + 1;
            queue.add(u.right);
        }
    }
    ArrayList<Integer> list = new ArrayList<>();
    for (int i = min; i <= max; i++) {
        if (!table.containsKey(i)) {
            continue;
        }
        ArrayList<Node> nodes = table.get(i);
        list.add(nodes.get(nodes.size() - 1).data);
    }
    return list;
}
```