```java
// Function to find the vertical order traversal of Binary Tree.
// Node{int data, Node left, Node right}
// Pair{Node data, int hd}
static ArrayList<Integer> verticalOrder(Node root) {
    Queue<Pair> queue = new LinkedList<>();
    queue.add(new Pair(root, 0));
    Hashtable<Integer, ArrayList<Node>> table = new Hashtable<>();
    int min = 9999999; // min horizontal distance from root
    int max = -9999999;// max horizontal distance from root
    while (!queue.isEmpty()) {
        Pair 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.node);
        } else {
            ArrayList<Node> list = new ArrayList<>();
            list.add(u.node);
            table.put(u.hd, list);
        }
        Node node = u.node;
        if (node.left != null) {
            queue.add(new Pair(node.left, u.hd - 1));
        }
        if (node.right != null) {
            queue.add(new Pair(node.right, u.hd + 1));
        }
    }
    ArrayList<Integer> list = new ArrayList<>();
    for (int i = min; i <= max; i++) {
        if (!table.containsKey(Integer.valueOf(i))) {
            continue;
        }
        for (Node u : table.get(i)) {
            list.add(u.data);
        }
    }
    return list;
}
```