**Bottom View Of Binary Tree**

vector<int> bottomView(BinaryTreeNode<int> \*root) {

    // Write your code here.

    vector<int> result;

    if (root == NULL)

        return result;

    map<int, int> bottomViewMap; // map to store the horizontal distance and node value

    queue<pair<BinaryTreeNode<int>\*, int>> q; // queue to perform level order traversal

    q.push(make\_pair(root, 0)); // insert root node with horizontal distance 0

    while (!q.empty()) {

        BinaryTreeNode<int>\* node = q.front().first;

        int hd = q.front().second;

        q.pop();

        // Update the horizontal distance with the latest node value

        bottomViewMap[hd] = node->data;

        // Process the left child

        if (node->left)

            q.push(make\_pair(node->left, hd - 1));

        // Process the right child

        if (node->right)

            q.push(make\_pair(node->right, hd + 1));

    }

    // Traverse the map and store the bottom view nodes

    for (auto it : bottomViewMap) {

        result.push\_back(it.second);

    }

    return result;

}