**Right View of a Binary Tree:**

class Node {

    constructor(val) {

      this.val = val;

      this.left = null;

      this.right = null;

    }

  }

  const a = new Node("a");

  const b = new Node("b");

  const c = new Node("c");

  const d = new Node("d");

  const e = new Node("e");

  const f = new Node("f");

  a.left = b;

  a.right = c;

  b.left = d;

  b.right = e;

  c.right = f;

  const one = new Node(1);

  const two = new Node(2);

  const three = new Node(3);

  const four = new Node(4);

  const five = new Node(5);

  const six = new Node(6);

  const seven = new Node(7);

  const eight = new Node(8);

  one.left = two;

  one.right = three;

  two.right = five;

  two.left = four;

  three.right = seven;

  three.left = six;

  four.right = eight;

function rightView(root){

  if (!root) return [];

  let result = [];

  let queue = [root];

  while(queue.length>0){

    let levelSize = queue.length

    for(let i = 0; i<levelSize; i++){

      let currentNode = queue.shift();

      if(i===levelSize -1){

        result.push(currentNode.val)

      }

      if(currentNode.left) queue.push(currentNode.left)

      if(currentNode.right) queue.push(currentNode.right)

    }

  }

  return result

}

console.log(rightView(a))

console.log(rightView(one))