**Finding Kth Smallest Element in a BST**

class Node {

  constructor(data) {

    this.data = data;

    this.right = null;

    this.left = null;

  }

}

class BST {

  constructor() {

    this.root = null;

  }

  insert(val) {

    const newNode = new Node(val);

    if (!this.root) {

      this.root = newNode;

      return;

    }

    let current = this.root;

    while (current) {

      if (val < current.data) {

        if (!current.left) {

          current.left = newNode;

          return;

        }

        current = current.left;

      } else if (val > current.data) {

        if (!current.right) {

          current.right = newNode;

          return;

        }

        current = current.right;

      } else {

        return;

      }

    }

  }

  printInOrder(node = this.root) {

    if (!node) return [];

    return [

      ...this.printInOrder(node.left),

      node.data,

      ...this.printInOrder(node.right),

    ];

  }

  printKthSmallest(node = this.root, k) {

    if (!node) return [];

    let arr = this.printInOrder(node);

    return arr[k];

  }

}

const tree = new BST();

tree.insert(5);

tree.insert(3);

tree.insert(7);

tree.insert(2);

tree.insert(4);

tree.insert(6);

tree.insert(8);

//console.log(tree.printInOrder())

console.log(tree.printKthSmallest(tree.root, 2));