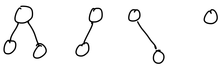
Bst

Friday, 11 November 2022 4:03 PM

what is BST?

① Should be a binary tree



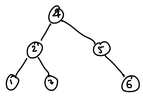
② Left subtree nodes < Root

③ Right subtree nodes > Root

④ Left & Right subtree are also BST with No dups.

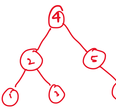
Inorder Traversal of BST is Sorted sequence.

L.S. → Root → R.S.



Inorder → 1, 2, 3, 4, 5, 6

BST Search

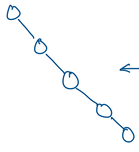


Binary Search → decide left? Right?

Same concept with BST.

average case is O(log N)

O(Height)



← Skewed Tree

O(Height)

=O(N)

---

strategy.

Problems will be solved using recursion;

Divide into subproblem & make recursive call on subtree

Code to create BST using array.

```
Class Node {
    Constructor (item) {
        this.key = item;
        this.left = null;
        this.right = null;
    }
}

var root = null;

function insertKey(key){
    root = insert(root, key);
}

function insert (key, root){
    if (root == null) {
        root = new Node(key);
        return root;
    }
    if (root.data > key){
        root.left = insert(root.left, key);
    }
    else {
        root.right = insert(root.right, key);
    }
    return root;
}
```

Code to search a Node

```
function search (root, key){
    if (root == null || root.data == key){
        return root;
    }
    if (root.data > key){
        return search(root.left, key);
    }
    else {
        return search(root.right, key);
    }
}
```

```
function isBST(root){
    return check(root, Number.MIN_VALUE, Number.MAXVALUE)
}

function check(node, min, max){
    if (root == NULL) return true;
    if (node.data < min || node.data > max){
        return false.
    }
    return check(node.left, min, node.data-1)
        && check(node.right, node.data+1, max);
}
```