

Trees



- Snarf the code for today's class
- and start looking at the code

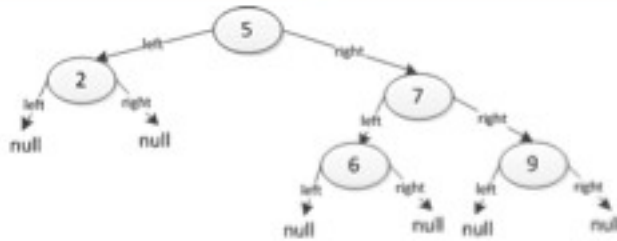


Today



- Binary search tree (BST) review
- Adding nodes to a BST
- Practice with BSTs and recursion

Binary Tree



```
IntTreeNode root = null;
```

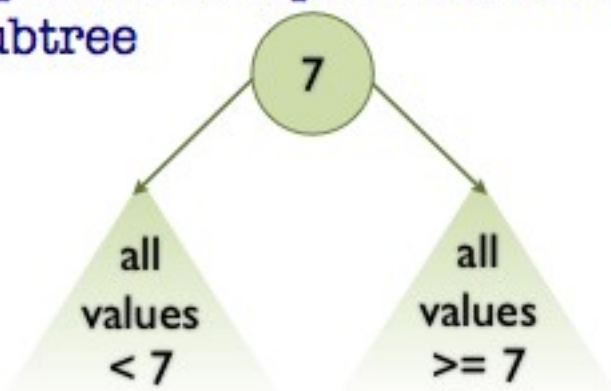
```
public class IntTreeNode {  
    public int value;  
    public IntTreeNode left; // holds smaller tree nodes  
    public IntTreeNode right; // holds larger tree nodes  
  
    public IntTreeNode(int val) { value = val; }  
}
```

3

Binary Search Tree



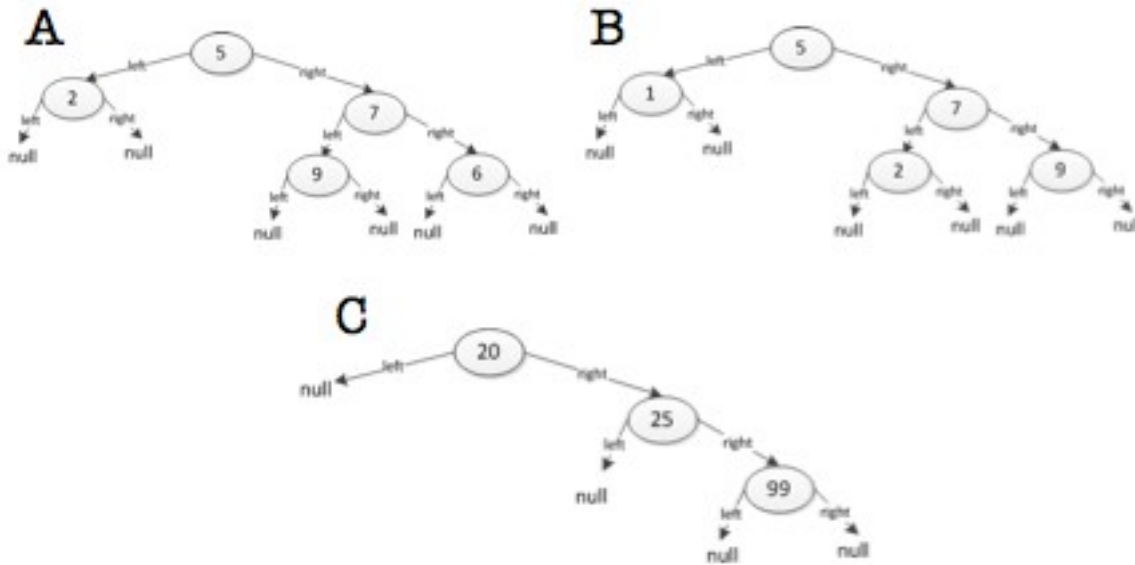
- Each node has a value
- Nodes with values **less than** their parent are in the **left** subtree
- Nodes with values **greater than, or equal to** their parent are in the **right** subtree



4



- Which is a binary search tree?



Question



- How long does it take to add **n** nodes into a binary search tree?

- Code time!

Question



- How long does it take to add n nodes into a binary search tree?
 - $O(\text{tree height})$
 - Complete binary tree: $O(\lg(N))$
 - Worst case: $O(N)$
 - (Amortized) Average: $O(\lg(N))$
- How do we keep the tree height small?

7

Amortized analysis

- The time required to perform a sequence of data-structure operations averaged over all the operations performed
- What do you need to know?

8

Code time



- Snarf the Recitation 8 code
- Complete the Recitation 8 assignment from the webpage