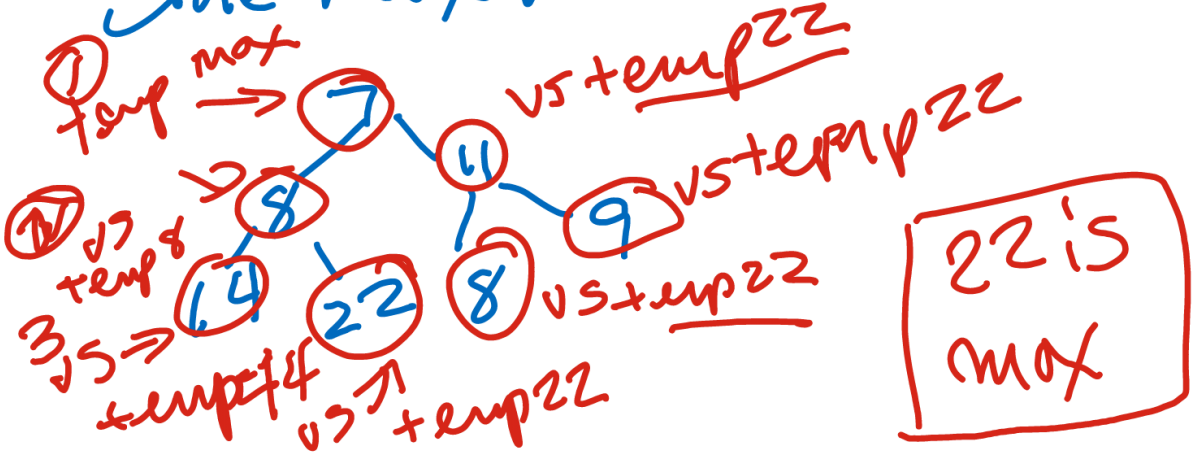
Whiteboard 18

WHITE BOARD 18

Find the maximum valve w/in a Binary tree of positive integers.

Visual

Traverse a binary tree and store & return the max valve



Algorithm

- est TreeNode root; new as property of class
- import Binary Tree package
- Create a recursive method pair
  public TreeNode
  private TreeNode (TreeNode)
  - in recursive private method

go to root/note
- compare value of each node to the max node value so far on the left
- repeat for the right

return TreeNode

Big O

~~O(2^n time~~

looked up on duke edu

~~O(log n)~~ time O^n per step

O(log n space per step

Pseudo Code

- after creating variables for TreeNode root & import Binary Tree package
- public method that only takes in TreeNode.root via private method
- private method takes in a Tree Node as current val

```
if(null return ∅)
While current.left &
  current.right !null
```

compare
current vs currentleft
w/ Math.max
. = currentmax

then compare
currentmax vs
current right
w/ Math.max
= currentmax

return currentmax
.data;

Tests

1. lop-sided Tree
2. Tree w/ dup values
3. Empty Tree?
4. Tree w/ only one node

Code

```
public Class FindMax {
    public TreeNode root;
```

```
public static TreeNode
    findMax() {
  return findMax(this.
      root)
}

private static TreeNode
  findMax (TreeNode node){
TreeNode max = 0;
TreeNode current = node;
if(current == null){
  return current;
}
while(current.left != null
  && current.right != null)
{
```

```
if (current.value >
current.left.value){
maxNode = current.
value;

}else if (current.value
< current.left
.value) {

maxNode =
current.left;

} else if(current.value
> current.right.value)
{maxNode =
current.value;

}
```

```
else if (current.value <
    current.right.value) {
    maxNode =
        current.right.value;
}
    }
    return MaxNode;
}
// endprogram
```