Quiz 5 Solution
Binary Tree Algorithms

```
<table>
<thead>
<tr>
<th>data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Left Subtree

Right Subtree
```
public int countGreater(int d) {
    //if the tree is empty return 0
    //otherwise return the number of values in the tree greater than d
    return countGreater(root, d);
}

private int countGreater(Node r, int d) {
    //if r is null return 0
    //otherwise return the number of values in the subtree rooted at r
    //that are greater than d
    if r is null return 0
    i = Find the number greater than d in the left tree
    j = Find the number greater that d in the right tree
    if the node is greater than d k = 1 else k = 0
    return i+j+k
};
public int countGreater(int d) {
    // if the tree is empty return 0
    // otherwise return the number of values in the tree greater than d
    return countGreater(root, d);
}

private int countGreater(Node r, int d) {
    // if r is null return 0
    // otherwise return the number of values in the subtree rooted at r
    // that are greater than d
    if (r == null) return 0;
    int i = r.data > d ? 1 : 0;
    return i + countGreater(r.left, d) + countGreater(r.right, d);
}
public int countLeaves() {
    //if the tree is empty return 0
    //otherwise return the number leaves in the tree
    return countLeaves(root);
}

private int countLeaves(Node r) {
    //if r is null return 0
    //otherwise return the number of leaves in the subtree rooted at r
    if (r is null return 0
    if r is a leaf return 1
    i = Find the number of leaves in the left tree
    j = Find the number of leaves in the right tree
    Return i+j;
}
public int countLeaves() {
    // if the tree is empty return 0
    // otherwise return the number leaves in the tree
    return countLeaves(root);
}

private int countLeaves(Node r) {
    // if r is null return 0
    // otherwise return the number of leaves in the subtree rooted at r
    if (r == null) return 0;
    if (r.left == null && r.right == null) return 1;
    return countLeaves(r.left) + countLeaves(r.right);
}