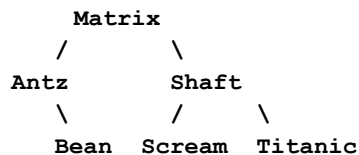


1. Is this a Binary Search Tree (yes or no)? NO (duplicate value) (1pt)
2. Is this a Binary Search Tree (yes or no)? NO (not by our definition of left trees < parents) (1pt)
3. What potential benefit is there to using a binary search tree rather than an array to store the same collection

Find, insert, and remove are all $O(\log n)$ whereas an array, even sorted, has $O(N)$ insert and remove

4. Draw the a picture of the binary search tree that results from the following code that uses our



5. // Add both methods below or on other paper

```

private int atDepth(int level) {
    return atDepth(root, level);
}

private int atDepth(TreeNode t, int level) {
    if (t == null)
        return 0;
    else if (level == 0)
        return 1;
    else
        return atDepth(t.left, level - 1) + atDepth(t.right, level - 1);
}

public boolean maxedOut() {
    int maxLevel = height(root);
    int nodesAtLevel = 1;
    for (int level = 0; level < maxLevel; level++) {
        nodesAtLevel *= 2;
        if (nodesAtLevel != atDepth(root, level))
            return false;
    }
    return true;
}

public int height(TreeNode t) {
    if (t == null)
        return -1;
    else
        return 1 + Math.max(height(t.left), height(t.right));
}
    
```