

Trees - Take Home Problems Solutions

1.

```
function isIdenticalTree(n1, n2) {
  if (!n1 && !n2) return true;
  if (!n1 || !n2 || n1.val !== n2.val) return false;
  return isIdenticalTree(n1.left, n1.left) && isIdenticalTree(n2.right,
n2.right);
}
```

2.

```
function height(root){
  if(root == null)
    return 0
  return Math.max(height(root.left), height(root.right)) + 1
}

function isBalanced(root){
  if(root == null)
    return true
  let lh = height(root.left)
  let rh = height(root.right)
  if (Math.abs(lh - rh) <= 1 && isBalanced(
root.left)== true && isBalanced( root.right) == true)
    return true
  return false
}
```

3.

```
var buildTree = function(preorder, inorder) {
  if (!preorder.length || !inorder.length) return null;
  let root = new TreeNode(preorder[0]);
  let rootIndexInorder = inorder.indexOf(preorder[0]);
  root.left = buildTree(preorder.slice(1,rootIndexInorder + 1),
inorder.slice(0,rootIndexInorder));
  root.right = buildTree(preorder.slice(rootIndexInorder + 1),
inorder.slice(rootIndexInorder + 1));
  return root;
}
```

4.

```
function lca(root, n1, n2){
  if(root == null ) return root
  if(root.data == n1 || root.data == n2 ){
    return root
  }
```

```
    }  
  
    let left = this.lca (root.left,n1,n2)  
    let right = this.lca (root.right,n1,n2)  
  
    if(left == null) return right  
    if(right == null ) return left  
  
    return root  
}
```