

Week 7 Section Solutions

1. Tree Traversal (traversalsB on CSBS)

pre-order	42 15 27 48 9 86 12 5 3 39
in-order	15 48 27 42 86 5 12 9 3 39
post-order	48 27 15 5 12 86 39 3 9 42

2. CountLeftNodes

```
int countLeftNodes(BinaryTreeNode* node) {
    if (node == nullptr) {
        return 0;
    } else if (node->left == nullptr) {
        return countLeftNodes(node->right);
    } else {
        return 1 + countLeftNodes(node->left) +
            countLeftNodes(node->right);
    }
}
```

3. isBST

```
bool isBSTHelper(BinaryTreeNode* node, BinaryTreeNode*& prev) {
    if (node == NULL) {
        return true;
    } else if (!isBSTHelper(node->left, prev) ||
        (prev && node->data <= prev->data)) {
        return false;
    } else {
        prev = node;
        return isBSTHelper(node->right, prev);
    }
}

bool isBST(BinaryTreeNode* node) {
    BinaryTreeNode* prev = NULL;
    return isBSTHelper(node, prev);
}
```

```

//alternate solution
bool isBSTHelper(TreeNode *node, int min, int max) {
    if(node == null) return true;
    if(node->data < min || node->data > max) return false;
    return isBSTHelper(node->left, min, node->data - 1) &&
        isBSTHelper(node->right, node->data + 1, max);
}

bool isBST(TreeNode *root) {
    return isBSTHelper(root, INT_MIN, INT_MAX)
}

```

4. swapChildrenAtLevel

```

void swapChildrenAtLevelHelper(BinaryTreeNode* node, int k, int
currentLevel) {
    if (node != nullptr) {
        if (currentLevel == k) {
            BinaryTreeNode* temp = node->left;
            node->left = node->right;
            node->right = temp;
        } else if (currentLevel < k) {
            swapChildrenAtLevelHelper(node->left, k,
                                     currentLevel + 1);
            swapChildrenAtLevelHelper(node->right, k,
                                     currentLevel + 1);
        }
    }
}

void swapChildrenAtLevel(BinaryTreeNode*& node, int k) {
    if (k <= 0) {
        throw k;
    }
    swapChildrenAtLevelHelper(node, k, 1);
}

```

5. Hash Functions

hash1 and hash4 are VALID.
hash4 is GOOD.

