Name: *(as it would appear on official course roster)* [redacted]	
Umail address: [redacted]@umail.ucsb.edu	section [redacted]
Optional: name you wish to be called if different from name above.	
Optional: name of "homework buddy" (leaving this blank signifies "I worked alone"	

1

h07

CS24 W19

h07: Chapter 10: Trees

ready?	assigned	due	points
true	Wed 02/20 02:00PM	Mon 02/25 09:00AM	40

You may collaborate on this homework with AT MOST one person, an optional "homework buddy".

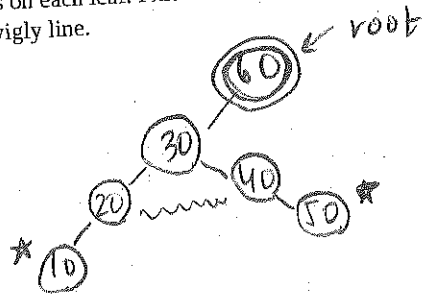
MAY ONLY BE TURNED IN IN THE LECTURE/LAB LISTED ABOVE AS THE DUE DATE, OR IF APPLICABLE, SUBMITTED ON GRADESCOPE. There is NO MAKEUP for missed assignments; in place of that, we drop the lowest scores (if you have zeros, those are the lowest scores.)

Read Chpater 10 (If you don't have a copy of the textbook yet, there is one on reserve at the library under "COMP000-STAFF - Permanent Reserve").

Please:

- **No Staples.**
- **No Paperclips.**
- **No folded down corners.**

1. (10 pts) Draw a binary search tree with 6 nodes containing the integer keys 60, 30, 40, 50, 20, 10. You can arrange these in any order as long as the resultant tree satisfies the properties of a BST. Circle the root, and put asterisks on each leaf. Find two nodes that are siblings and connect them with a wigly line.



2. (10 pts) Look at the tree in Figure 10.5 on page 484. In what order are the letters printed for an in-order traversal? What about a post-order traversal?

In-order: O L R A G T

Post-order: O R L T G A

2

h07

CS24 W19

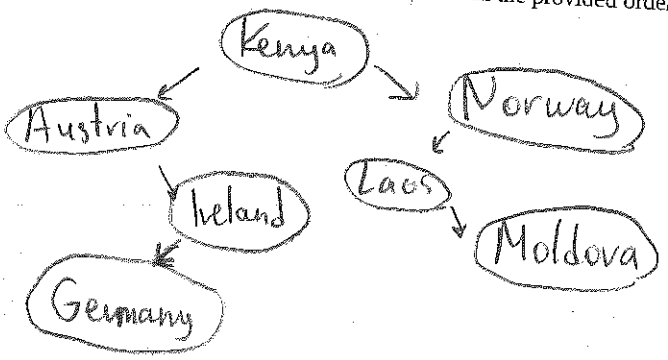
3. (10 pts) Write a function with the prototype void stars(int i);. The function prints a line of i stars followed by a new line. Write code to apply the stars function to every node in a binary tree using an in-order traversal, where the number of stars is that given by the data of the node in the BST. You must write both the function definition and the code that uses it with the star function

```
void stars(int i){
    string line = "";
    for(int j=0; j<i; j++) line += "*";
    cout << line << endl;
}

void BST::starAllHelper(){
    starAll(this->root);
}

void BST::starAll(Node* n)
    if(n){
        starAll(n->left);
        stars(n->info);
        starAll(n->right);
    }
```

4. (2 pts) Why is it bad to insert nodes from smallest to largest in a binary search tree?

Because you are not taking advantage of BST's advantages in searching, etc. Your BST will look like a linked list.

5. (5 pts) Build a binary search tree with the following words (inserted in the provided order): Kenya, Austria, Ireland, Norway, Laos, Germany, Moldova



6. (3 pts) Search for the following keys in the BST that you created in the previous question: Ireland, Germany, Moldova and in each case indicate the nodes that you visited along the way.

Ireland: Kenya → Austria → Ireland

Germany: Kenya → Austria → Ireland → Germany

Moldova: Kenya → Norway → Laos → Moldova