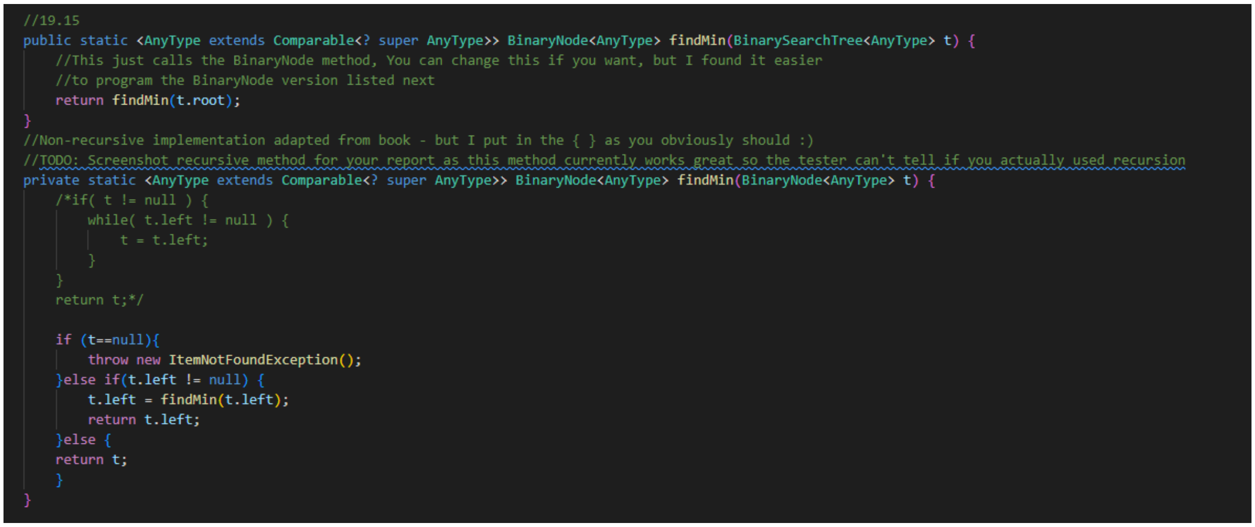
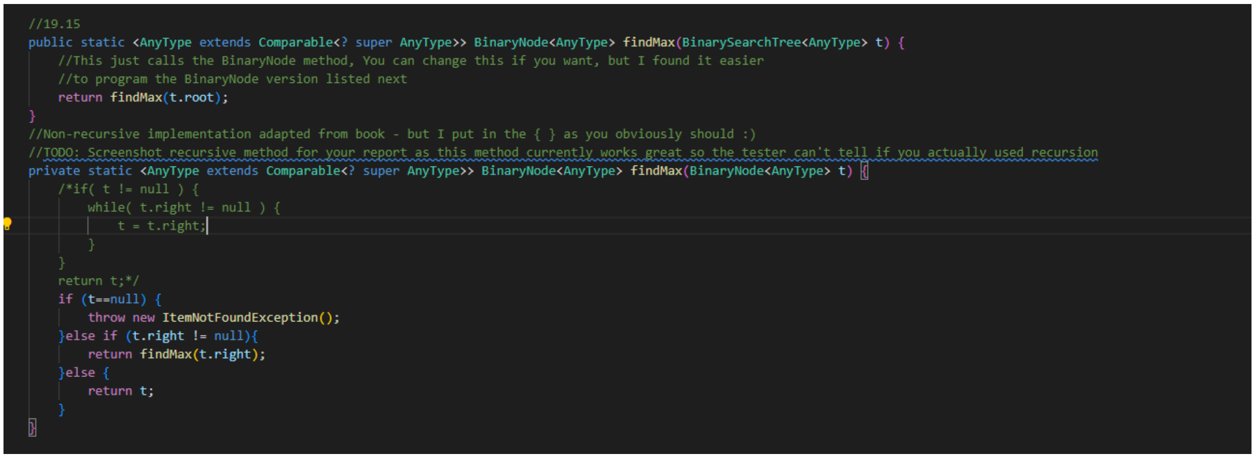
Chapter 19 Programming

19.15 The findMin method was pretty easy to figure out. All I did was run down the left side until I found a null



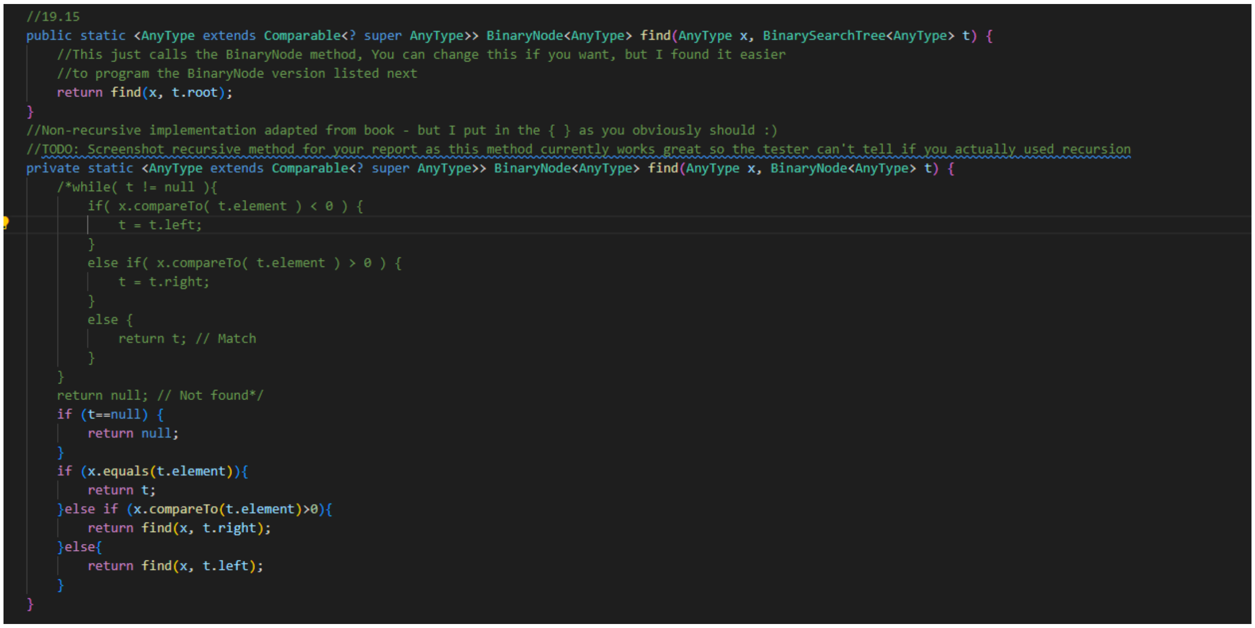
```
//19.15
public static <AnyType extends Comparable<? super AnyType>> BinaryNode<AnyType> findMin(BinarySearchTree<AnyType> t) {
    //This just calls the BinaryNode method, You can change this if you want, but I found it easier
    //to program the BinaryNode version listed next
    return findMin(t.root);
}
//Non-recursive implementation adapted from book - but I put in the { } as you obviously should :)
//TODO: Screenshot recursive method for your report as this method currently works great so the tester can't tell if you actually used recursion
private static <AnyType extends Comparable<? super AnyType>> BinaryNode<AnyType> findMin(BinaryNode<AnyType> t) {
    /*if( t != null ) {
        while( t.left != null ) {
            t = t.left;
        }
    }
    return t;*/

    if (t==null){
        throw new ItemNotFoundException();
    }else if(t.left != null) {
        t.left = findMin(t.left);
        return t.left;
    }else {
    return t;
    }
}
```

The findMax method was the exact same as the findMin but I just went to the right instead of the left.



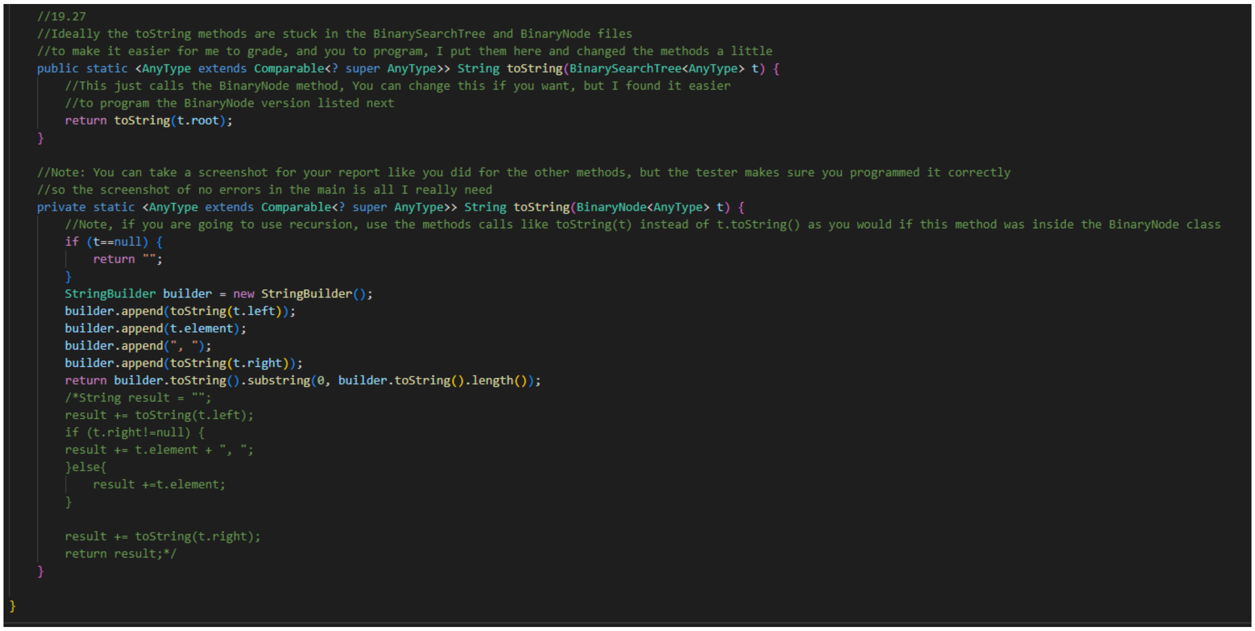
```
//19.15
public static <AnyType extends Comparable<? super AnyType>> BinaryNode<AnyType> findMax(BinarySearchTree<AnyType> t) {
    //This just calls the BinaryNode method, You can change this if you want, but I found it easier
    //to program the BinaryNode version listed next
    return findMax(t.root);
}
//Non-recursive implementation adapted from book - but I put in the { } as you obviously should :)
//TODO: Screenshot recursive method for your report as this method currently works great so the tester can't tell if you actually used recursion
private static <AnyType extends Comparable<? super AnyType>> BinaryNode<AnyType> findMax(BinaryNode<AnyType> t) {
    /*if( t != null ) {
        while( t.right != null ) {
            t = t.right;
        }
    }
    return t;*/
    if (t==null) {
        throw new ItemNotFoundException();
    }else if (t.right != null){
        return findMax(t.right);
    }else {
        return t;
    }
}
```

The find method was a little tougher and I still get an error with the tester. I am not sure what is going on with this because the provided iterative code also gets the same error. So either the tester is messed up or the provided code is.



```
//19.15
public static <AnyType extends Comparable<? super AnyType>> BinaryNode<AnyType> find(AnyType x, BinarySearchTree<AnyType> t) {
    //This just calls the BinaryNode method, You can change this if you want, but I found it easier
    //to program the BinaryNode version listed next
    return find(x, t.root);
}
//Non-recursive implementation adapted from book - but I put in the { } as you obviously should :)
//TODO: Screenshot recursive method for your report as this method currently works great so the tester can't tell if you actually used recursion
private static <AnyType extends Comparable<? super AnyType>> BinaryNode<AnyType> find(AnyType x, BinaryNode<AnyType> t) {
    /*while( t != null ){
        if( x.compareTo( t.element ) < 0 ) {
            t = t.left;
        }
        else if( x.compareTo( t.element ) > 0 ) {
            t = t.right;
        }
        else {
            return t; // Match
        }
    }
    return null; // Not found*/
    if (t==null) {
        return null;
    }
    if (x.equals(t.element)){
        return t;
    }else if (x.compareTo(t.element)>0){
        return find(x, t.right);
    }else{
        return find(x, t.left);
    }
}
```

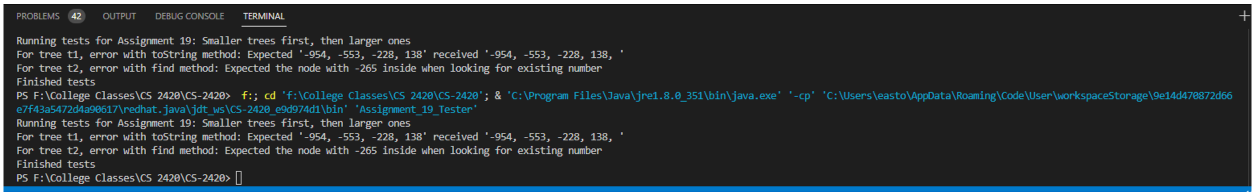
19.27 The toString method, I could not figure out the formatting but other than that, the method works perfect. I am going to the tutoring center tomorrow to solve this. I tried creating the method two different ways and both had the same issues.



```
    //19.27
    //Ideally the toString methods are stuck in the BinarySearchTree and BinaryNode files
    //to make it easier for me to grade, and you to program, I put them here and changed the methods a little
    public static <AnyType extends Comparable<? super AnyType>> String toString(BinarySearchTree<AnyType> t) {
        //This just calls the BinaryNode method, You can change this if you want, but I found it easier
        //to program the BinaryNode version listed next
        return toString(t.root);
    }

    //Note: You can take a screenshot for your report like you did for the other methods, but the tester makes sure you programmed it correctly
    //so the screenshot of no errors in the main is all I really need
    private static <AnyType extends Comparable<? super AnyType>> String toString(BinaryNode<AnyType> t) {
        //Note, if you are going to use recursion, use the methods calls like toString(t) instead of t.toString() as you would if this method was inside the BinaryNode class
        if (t==null) {
            return "";
        }
        StringBuilder builder = new StringBuilder();
        builder.append(toString(t.left));
        builder.append(t.element);
        builder.append(", ");
        builder.append(toString(t.right));
        return builder.toString().substring(0, builder.toString().length());
        /*String result = "";
        result += toString(t.left);
        if (t.right!=null) {
        result += t.element + ", ";
        }else{
            result +=t.element;
        }

        result += toString(t.right);
        return result;*/
    }

}
```

Assignment 19 Tester

Here is what the tester returns.



```
Running tests for Assignment 19: Smaller trees first, then larger ones
For tree t1, error with toString method: Expected '-954, -553, -228, 138' received '-954, -553, -228, 138, '
For tree t2, error with find method: Expected the node with -265 inside when looking for existing number
Finished tests
PS F:\College Classes\CS 2420\CS-2420>  f:; cd 'f:\College Classes\CS 2420\CS-2420'; & 'C:\Program Files\Java\jre1.8.0_351\bin\java.exe' '-cp' 'C:\Users\easto\AppData\Roaming\Code\User\workspaceStorage\9e14d470872d66
e7f43a5472d4a90617\redhat.java\jdt_ws\CS-2420_e9d974d1\bin' 'Assignment_19_Tester'
Running tests for Assignment 19: Smaller trees first, then larger ones
For tree t1, error with toString method: Expected '-954, -553, -228, 138' received '-954, -553, -228, 138, '
For tree t2, error with find method: Expected the node with -265 inside when looking for existing number
Finished tests
PS F:\College Classes\CS 2420\CS-2420> 
```