CS211 ALGORITHMS & DATA STRUCTURES II

LAB 4

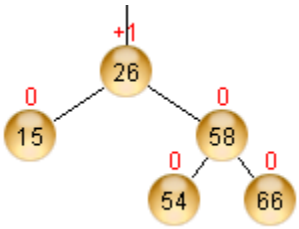
Dr. Phil Maguire

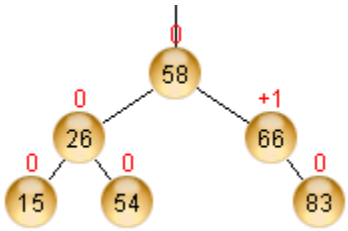
BALANCED BINARY TREES

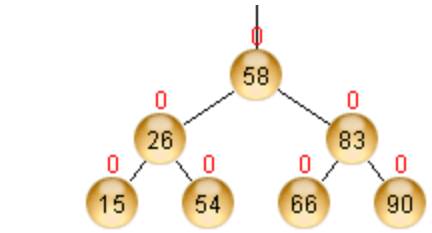
Pen and Paper Exercise

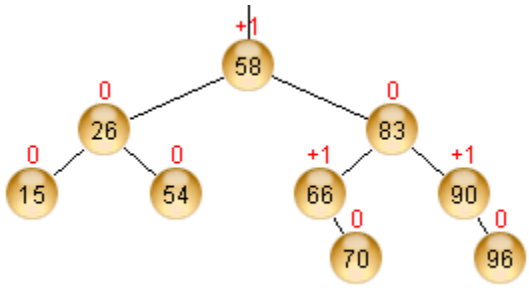
Add the following numbers into an AVL tree. To make sure you get it right you should show clearly the balance factors for each node and the rotations involved.

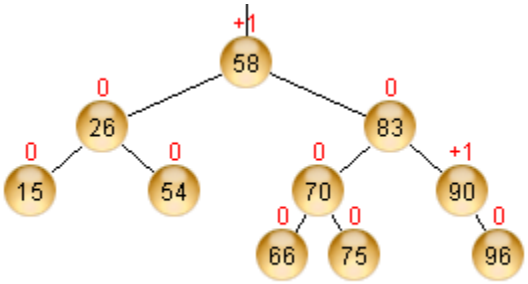
26 15 54 66 58 83 90 70 96 75 72

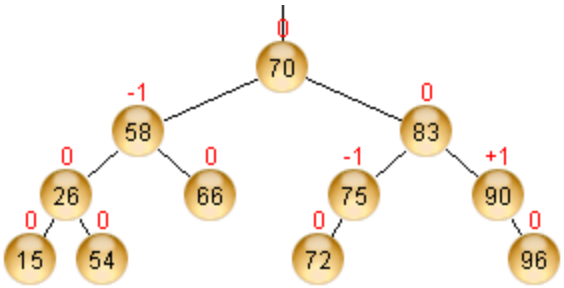












Programming Exercise

```
//this program outputs all the words of equal longest length that can
be formed with the selection of letters

import java.util.*;

public class CountDown{

    public static void main(String[] args){
```

```
        FileIO reader = new FileIO();
        String[] contents = reader.load("C:\\dictionary.txt");

        Scanner myScanner = new Scanner(System.in);
        System.out.println("Enter the string of random letters");
        String input = myScanner.nextLine();
        int biggest = 0;
        ArrayList<String> myList = new ArrayList<String>();
        for (int i = 0; i< contents.length; i++){
            int length =
check(input,contents[i].substring(0,contents[i].length()-1));

//the reason for the substring above is that each entry in the
contents array seems to include the return character and we don't want
to count this

            if(length>biggest){
                biggest=length;
                myList.clear();
                myList.add(contents[i]);
            }else if(length==biggest){
                myList.add(contents[i]);
            }
        }
        System.out.println("The longest word that can be formed is one
with "+biggest+" letters:");
        for(int i=0;i<myList.size();i++){
            System.out.println(myList.get(i));
        }
    }

    public static int check(String letters, String word){

        for(int i=0; i<word.length(); i++){
            if(!letters.contains(""+word.charAt(i))){
                return -1;
            }
            letters=letters.replaceFirst(""+word.charAt(i)," ");
        }
        return word.length();
    }
}
```