

Due September 4th, 2013

1. Programming (50 points)

Write a program in C++ called **homework1** that reads a list of positive and negative integers from user input (i.e. standard input) until a “END” statement is entered. The program should not display any text prompting the user for input. It should just start reading from standard input.

Organize these integers into a binary search tree. The first integer will be the root, the second integer should be inserted left, if smaller than the root, or right if larger.

After all numbers are inserted, the program should display the maximum height of the tree. The root node will be at level 0. Your program should also display the maximum width of the tree. This is the total number of nodes at any level. It should also display the level of the maximum width.

Your program’s output should look exactly like this (gray indicates command line):

```
yourName@elec>make clean
yourName@elec>make all
yourName@elec>./homework1 < testFile.txt
Tree Height: 10
Maximum Width: 4
Maximum Width Level: 4
```

If in fact the tree height is 10, it’s maximum width is 4, and the maximum width level is 4. You are required to use a binary search tree. You should use the depth first approach discussed in class for the tree height, and a breadth first approach for the tree width using a queue also discussed in class.

Grading Criteria

Construction of Data structures

Implementation

Correctness (includes not crashing and the output format meets the above criteria)

You need to submit any source code required to compile and run this assignment. You must use make as described in the syllabus. If the program does not compile, no credit is awarded. The homework will be tested on *elec.tricity.wsu.edu* and you should use the g++ compiler.

Note: above I redirect standard input from a text file using `< testFile.txt`. You should test your software with different test file trees. Consider discussions from the first day of class to test balanced and unbalanced trees. Also, you are allowed to use the standard template library (std namespace: cin, cout, endl) including `<string>` Refer to Koenig and Moo for potential C++ assistance. Or use `scanf`, `strcmp`, and `atoi`.

2. 10 points - Write functions for the below time complexities in pseudo code that accepts an integer element, k , and an integer array X .
- $O(N)$
 - $O(\lg N)$

What, if any, assumptions should you make about the format of the array when writing the $O(\lg N)$ search?

3. 10 points - Write pseudo code to search a binary search tree for the below time complexities:
- $O(\lg N)$
 - $O(N)$

You are allowed to use recursion. You should include ***typedefs*** for any data structures.

Why does your solution for part b run in $O(N)$ time?

4. 10 points - Write pseudo code to reverse an array using a stack. You must define the operations of a stack (push and pop). You should include ***typedefs*** for any data structures.
5. 10 points – Simple Sorting Methods
- What is the time complexity for bubble sort?
 - Write pseudo code for the bubble sort algorithm on an array X .
 - Write pseudo code for sorting array X using a binary search tree.
 - What is the time complexity for this solution?