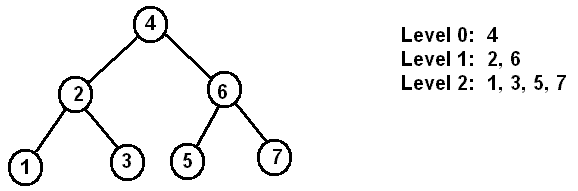
EECS 560 Spring 2016 Lab 2

This lab is due by midnight two days after your lab day. For example, Monday’s lab is due by Wednesday midnight (i.e. 12:00 a.m. Thursday). Thursday’s lab is due by midnight on Saturday.

Although we have not yet talked about trees in class, this lab only requires the knowledge you gained about binary search trees when you took EECS 268.

Problem: Given a sorted list of integers, build an optimal binary search tree—a binary search tree in which all searches can be done in O(lg n) time. The algorithm to build the tree should run in O(n) time. You will also have to do a level-order traversal of the tree you built in order to print out your results. Suppose you are given the list 1 2 3 4 5 6 7. Then you would construct the tree below and write out the result as indicated.



A level order traversal as the name indicates writes out the nodes of the tree one level at a time. The root is on level 0. You will need a queue to implement the traversal. You are to thoroughly test your program and, in addition to the code, turn in any test data you used and the results produced by your program on that data. Your program will be run on a different data set to ensure that it works correctly.

Here’s a basic level order traversal. Modify it as needed so that you can also print out the level numbers. r is the root of the tree and Q is a queue

**level\_order(r, Q)**

**initialize Q to contain the root r**

**while Q is not empty do begin**

**v** ← **dequeue(Q) // Remove head of queue**

**put the left and right children of v on the queue // if one child is null ignore it**

**end // while loop**

Specifications:

Your program should be designed to read the integers to be put into the tree from a file called data2.txt. Read the values in one at a time from the input stream—i.e. do not read the values into a string and then extract them from that string. You may assume that the file will contain at most 100 elements so you may statically allocate an array of that size.