CSE 331 Homework 4 (Due by Thursday's class on 10/17)

Programming part is due by Friday on 10/18

1. Prove that an n-element heap has height $h = \lfloor \log_2 n \rfloor$. You need to prove this for any given n. Showing $h = 3$ for $n = 8$ is not a proof. (5 points)

2. Apply ExtractMIN to the heap H[ ] = {2, 3, 7, 4, 9, 8, 15, 13, 10} using the pseudocode in lect11 (http://www.cse.msu.edu/∼ cse331/). Show the contents of the **array** H whenever there is a change. (3 points)

3. Insert key 1 to the following heap H using the pseudocode in lect11 (http://www.cse.msu.edu/∼ cse331/). Show the contents of the **array** H whenever there is a change. H[ ] = {2, 3, 7, 4, 9, 8, 15, 13, 10}. (4 points)

4. This problem consists of two parts. *First*, give an $O(n \log_2 k)$-time algorithm to merge k sorted lists into one sorted list, where n is the total number of elements in all the input lists. (*Hint*: Use a min-heap for k-way merging.) Note, if you need to use any defined operation on a heap, just treat it as a function and don't need to worry about the details. For example, if you want to build a heap on an array A, just say buildHeap(A). However, you need to describe on what elements you build this heap. For your designed algorithm, *analyze its time complexity*.

   In the *second* part of this problem, show the contents and changes of the heap you built in order to merge three sorted lists $[0, 1, 3]$, $[4]$, and $[2, 5]$. (15 pts)

5. This problem is a programming problem. Implement a program which builds a heap for any input array using the algorithm described in our class (see Chapter 7 or the notes). A test program is provided at

http : *//www.cse.msu.edu/* ∼ *cse*331/*examples/heap* − *shell.cpp.*

Only submit your **source codes and a readme.txt file** that explains how to compile and run your program. We will compile your program on arctic.cse.msu.edu. Please make sure it can compile and run correctly on arctic. For any readme file, please use .txt format. **Don't use doc, pdf, ps, zip, docx.** 10 pts for compile. 20 pts for correct implementation. (30 pts)