CS 430 – Fall 2016 – Sixth Quiz

Solve ONLY the third problem of this previously-given midterm. Submit your work on separate paper (make sure you add your name), and keep this sheet. This is an individual quiz, and no electronic help is allowed. Everything with an on/off button must be off. Books and notes are allowed. You have 30 minutes. Good luck!

"This exam is closed books and closed notes. Time: 1h 15m"

You can use any procedure or algorithm discussed in class (such as MergeSort() or ReheapDown()), but you must give complete specifications and state the running time of the procedure in terms of its parameters. For example, you can use:

```
PARTITION(A,p,q,r)
Input: Array A, indices p and q
Output: By swapping elements of A[p...q], we have that A[r] is the initial A[p] (that is, A[p] is the pivot), all elements of A[p...r−1] are at most A[r], and all elements of A[r+1...q] are at least A[r],
Running time: O(q−p)
```

To help me grade, please respect the following guidelines for writing pseudocode:

1. C instructions are fine. But do not write object-oriented additions. Do not declare or use any class. Declare only procedures (if necessary) and explain in words what each procedure does, and what is the use of each parameter.

2. One instruction per line

3. from C, do not use ‘!’, ‘:’, ‘?’, or double assignment. As a general rule, this is not the place to be too smart

4. Match the brackets with a horizontal line

5. Number the lines of pseudocode

6. By default, the arrays are indexed 0...n−1. If you prefer to use the range 1...n, it is OK, but you must write down this assumption.

7. It can help the grader if you describe in English the idea of the algorithm, or examples of instances on which you run your algorithm. This can help your grade if your idea is correct but the pseudocode is wrong.

**Problem 1** (20 pts)

For this problem, assume that heaps have maximum in the root. Is the following array a heap:

<23, 17, 14, 13, 16, 10, 1, 5, 7, 12>?

Please explain your answer below.

**Problem 2** (20 pts)

Order the following list of functions by the big-Oh notation. Group together (for example, by underlining) those functions that are big-Theta of one another.
\[
\begin{align*}
&n^{2.5} & n! \\
n \log n & (\log n)^4 \\
3^n & n^2 \log n
\end{align*}
\]

**Problem 3** (30 pts)

Suppose you are given \( k \) \( n \)-element sorted sequences \( A_i \), each representing a set (none has duplicate entries). Describe an \( O(nk \log k) \)-time method for computing a sorted sequence representing the set \( \bigcup_{i=1}^{k} A_i \) (with no duplicates).

You do not have to argue correctness (but, obviously, your method must be correct), but must justify the running time.

**Hint:** use a data structure from the class and a framework as in a homework. State the operations the data structure supports, what are their parameters, what is their results, and with what running time. Copy your pseudocode here.

**Problem 4** (30 pts)

In the **art gallery guarding** problem we are given a line \( L \) that represent a long hallway in an art gallery. We are also given a set \( X = \{x_0, x_1, \ldots, x_{n-1}\} \) of real numbers that specify the position of paintings in this hallway. Suppose a single guard can protect all the paintings within distance 1 of his position (on both sides).

Design an algorithm for finding a placement of guards that uses the minimum number of guards to protect all the paintings with positions in \( X \). Present the pseudocode. Prove that your algorithm minimizes the number of guards, and analyze its running time which must be polynomial; \( O(n \log n) \) gives an extra 3 points.