Fall 2009: CS 430 - Midterm solution sketches

Problem 1 For this problem, assume that heaps have maximum in the root. Is the following array a heap: $< 23, 17, 14, 13, 16, 10, 1, 5, 7, 12 >$?

Yes, this is a max heap. For each node in the heap it is a leaf or it is larger than its children.

Problem 2 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.

$n^{2.5}$	$n!$
$n \log n$	$(\log n)^4$
3^n	$n^2 \log n$

Ordered from slowest growing to fastest growing: $(\log n)^4, n \log n, n^2 \log n, n^{2.5}, 3^n, n!$

Problem 3 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 sequence representing the set $\cup_{i=1}^{k} A_i$ (with no duplicates).

```
    UNION(A, k)
1   let U be the union of sequences all of the sequences A_i
2   build a min heap with k nodes (one for each sequence)
(The initial values of the nodes are the first values in each sequence)
3   while the heap is not empty do
4     get the min value in the heap
5     if the min value does not match the last value in U
6       then append this value to U
7     get the next value in the sequence for this node and update the heap.
8     if the sequence does not have anymore values, then remove its node from the heap.
9   return U
```

Building the heap takes $O(k)$ time. The while loop will execute $O(nk)$ times. Each iteration of the while loop takes $O(\log k)$ time to update the heap. The total runtime is $O(nk \log k)$.

Problem 4 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.

```
    PLACE-GUARDS(X)
1   sort the values in X in ascending order
2   nGuards ← 1
3   gPos[1] ← X[0] + 1
4   for i ← 1 to n-1
5     if X[i] is not protected by the guard at gPos[nGuards]
6       then nGuards ← nGuards + 1
7       gPos[nGuards] ← X[i] + 1
8   return {nGuards, gPos}
```

Sorting X takes $O(n \log n)$ time. Placing the guards takes $O(n)$ time. The total runtime is $O(n \log n)$.