CS 573: Graduate Algorithms, Fall 2011
HW 0 (due Tuesday, August 30th in class)

This homework contains four problems. **Read the instructions for submitting homework on the course webpage**. In particular, *make sure* that you write the solutions for the problems on separate sheets of paper. Write your name and netid on each sheet.

Collaboration Policy: For this home work, each student should work *independently* and write up their own solutions and submit them.

0. Write the sentence ***"I understand the course policies. I worked on this homework on my own."*** Solutions that omit this sentence will not be graded.

1. (10 pts) Sort the following functions from asymptotically smallest to asymptotically largest. Your answer should be a sorted list. Note that there may be ties.

$$\begin{array}{cccc} \ln\ln n & (\sqrt{3})^{\lg n} & n^{1+1/\lg n} & e^{\ln\ln n} \\ \sqrt{n} & H_n & 4^{H_n} & n^{1.2} \\ \lg^{\lg n} n & n^n & n! & (1+1/n)^{3n\ln n} \end{array}$$

   To jog your memory: $\lg n = \log_2 n$ and $\ln n = \log_e n$ and $H_n = \sum_{i=1}^{n} 1/i \simeq \ln n + 0.577215\ldots$ is the n'th harmonic number.

2. (15 pts) Solve the following recurrences in the sense of giving an asymptotically tight bound of the form $\Theta(f(n))$ where $f(n)$ is a standard and well-known function. No proof necessary.

   (a) $A(n) = n^{1/3}A(n^{2/3}) + n$, $A(n) = 1$ for $1 \le n \le 8$.

   (b) $B(n) = B(n/2) + \sqrt{n}$, $B(1) = 1$.

   (c) $C(n) = 3C(n-1) + 1$, $C(1) = 0$.

   (d) $D(n) = D(\log n) + \log n$, $D(1) = 0$.

   (e) $E(n) = 3E(n/3) + 4E(n/4) + n^3$, $E(n) = 1$ for $n \le 4$.

3. (25 pts) A standard priority queue data structure stores keys that have associated priorities and supports two basic operations: insertion of a key with a given priority, and extracting a key with the highest priority. Priority queues can be implemented at the cost of $O(\log n)$ per operation where n is the number of keys stored in the data structure. In some applications the number of distinct priorities, k, is often very small compared to the number of keys and the parameter k is often known in advance. Design a modified priority queue data structure that supports insertions and extract in $O(\log k)$ time per operation. Your data structure knows k in advance and should report an error if more than k distinct priorities are inserted into it at any point. You should assume that when exctracting a key with highest priority it does not matter which of the keys with that priority is returned. *Hint: combine a standard priority queue with an auxiliary standard data structure.*

4. (25 pts) Let $G = (V, E)$ be an undirected and connected graph. Fix a vertex $u \in V$. Let $\mathcal{T}_1$ be the set of all spanning trees that can be obtained by doing a depth first search (DFS) in G starting with u. Note that there are multiple DFS trees possible because of the choice available in exploring the neighbors of a node. And let $\mathcal{T}_2$ be the set of all spanning trees that can be obtained by doing a breadth first search (BFS) starting with u. Prove that there are trees $T_1 \in \mathcal{T}_1$ and $T_2 \in \mathcal{T}_2$ such that $T_1 = T_2$ (in the sense that they have the same set of edges) if and only if G is a tree. *Hint: Understand how DFS and BFS explore the graph if it is a simple cycle.*

5. (25 pts) Consider the standard balls and bins process. A collection of m identical balls are thrown into n bins: each ball is thrown independently into a bin chosen uniformly at random.

   (a) (5 pts) What is the (precise) probability that a particular bin i contains exactly k balls at the end of the experiment? Let X be the (random) number of bins that contain exactly k balls. What is the expected value of X?

   (b) (10 pts) What is the variance of X?

   (c) (10 pts) Consider the same experiment but with $m = n$. Now balls and bins are numbered from 1 to n. We say that there is a match in bin i if at the end of the experiment it contains exacly the ball numbered i. What is the probability that there are exactly k matches?