CIS 435: Homework 2 (Due: February 10, 2004)

Solve problems 1-3 (Group 1) and either Problem 4 (Group 2) or Problem 5 (Group 3).

Problem 1. (7 points)

Rank the following functions by order of growth; that is, find an arrangement $g_1, g_2, g_3, \ldots g_{14}$ of the functions satisfying $g_1 = \Omega(g_2), g_2 = \Omega(g_3), g_3 = \Omega(g_4), \ldots, g_{13} = \Omega(g_{14})$. Partition your list in equivalence classes such that $g(n)$ and $h(n)$ are in the same class if and only if $f(n) = \Theta(g(n))$.

$$3^n,\ n^4,\ n^{\lg\lg n},\ 8^{\lg n},\ (\lg n)^{\lg n},\ n^2,\ 2^{4\lg n},\ \lg(n!),\ 2^{\lg n},\ n^n,\ n\lg n,\ (1/2)^n,\ 1,\ (n-1)!,$$

Problem 2. (6 points)

Replace ? with each one of $\omega, o, \Theta, O, \Omega$ that makes the relationship between the two sides true. If none is applicable write "none". If necessary, use the approximation $n! \approx (n/e)^n$, where $e = 2.71\ldots$. Assume that $k \geq 1$, $\epsilon > 0$ and $c > 1$ are constants. Justify your answers.

1. $\lg(n!) =\ ?\ (2^n)$.
2. $2^n =\ ?\ (4^{n/2})$.
3. $n^{2\lg n} =\ ?\ (4^{\lg n * \lg n})$.
4. $1 =\ ?\ (1/n)$.
5. $n+1 =\ ?\ (n-1)$.
6. $16^n =\ ?\ (4^{2n+c})$.

Problem 3. (7 points)

Describe an $\Theta(n\lg n)$-time algorithm that, given a set S of n integers and another integer x, determines whether or not there exist two elements a, b in S whose sum $a+b$ is exactly x.

Problem 4. (30 points)

(a) Solve the following recurrences. You may assume $T(1) = \Theta(1)$, where no boundary condition is given. Make your bounds as tight as possible. Use asymptotic notation to express your answers. Justify your answers. (12 points)

i. $T(n) = 25T(n/5) + n^2$.
ii. $T(n) = 27T(n/3) + n^4$.
iii. $T(n) = 9T(n/3) + n$.

(b) Consider a modification of merge-sort in which out of n keys n/k subsequences of length k are formed and each one is sorted by insertion sort and then all subsequences are merged. The optimal k will be determined after analyzing the worst case performance of the algorithm. In other words, this new algorithm works as follows. (8 points)

(i) Form n/k subsequences each of length k. Sort each one with insertion sort. What is the total (worst-case) running time of all k subsequences using insertion sort in asymptotic notation? Explain.

(ii) The k subsequences can be merged in $\Theta(n\lg(n/k))$ time. Show how.

(iii) Show that the (worst-case) running time of the whole algorithm is $O(nk + n\lg(n/k))$.

(iv) What is the (asymptotically) largest value of k for which this algorithm is as good as merge-sort? Explain.

(c) We have 243 coins all of the same weight except one that is a fake and weighs less than the others. We also have a balance scale; any number of coins can be put on one or the other side of the scale at any one time and the scale will tell us whether the two sides weigh the same or which side weighs more (or less). Can you find the fake coin with only 5 weighings? Explain (and justify your answer). (10 points)

Problem 5. (30 points)

Do the Programming Module that is outlined in the electronic handout available at the course web page
`http://www.cs.njit.edu/~alexg/courses/cis435/handouts/phw2.ps`
or
`http://www.cs.njit.edu/~alexg/courses/cis435/handouts/phw2.pdf`