Qualifying Exam: Spring 2006
Data structures, algorithms, and complexity

Do all problems. No books or notes may be used. Points will be given based on the correctness, completeness, and style of your answers. Each problem is worth 20 points.

Problem 1:

a). (5 points): Define precisely what $f(n) = O(g(n))$, $f(n) = \Omega(g(n))$, and $f(n) = \Theta(n)$ mean.

b). (15 points): Prove the following or give a counterexample:

If $f_1(n) = O(g_1(n))$ and $f_2(n) = O(g_2(n))$, then $f_1(n) \cdot f_2(n) = O(g_1(n) \cdot g_2(n))$.

Note: this question is slightly modified from how it was actually asked when the exam was given. The question shown here was the intended question.

Problem 2:

Suppose you are given an array A of n sorted numbers that has been *circularly shifted* k positions to the right. For example, $\{35, 42, 5, 15, 27, 29\}$ is a sorted array that has been circularly shifted $k = 2$ positions, while $\{27, 29, 35, 42, 5, 15\}$ has been shifted $k = 4$ positions.

a). (5 points): Suppose you know what k is. Given an $O(1)$ algorithm to find the largest number in A.

b). (15 points): Suppose you *do not* know what k is. Give an $O(\log n)$ algorithm to find the largest number in A.

Problem 3:

a). (5 points): Define NP.

b). (5 points): Define NP-complete.

c). (10 points): Describe, at a high level, how it is proven that SAT is NP-complete without assuming the NP-completeness of any other language. (You may give a detailed proof instead if you prefer, but this is not necessary.)