CS 410 Summer 2000
Clarifications for Homework 1

Question 1. There are several clarifications for this question:

(a) For part (a), think about this: can either bubble sort or selection sort easily be made more efficient for the best case? Hint: use a flag.

(b) Finding the smallest value should not involve modifying either array, although of course putting that smallest value somewhere may.

(c) It's OK to modify array A for the initial problem here. In fact, if it's more natural for you to give pseudocode for part c, that's fine. Your answer for part c will be something like "see pseudocode above".

(d) In part (c), you need only describe the changes necessary, you do *not* need to give full pseudocode.

Question 2. Find the smallest ***integer*** n.
One way to show your answer is correct is to show the appropriate values for two consecutive integers.

Question 4. min and max of functions are computed at each point. For example, in the expression $O(min(f(n), g(n)))$, for each n the value of $f(n)$ and the value of $g(n)$ is computed, and the lower of the two values is selected.

Question 8. There are two clarifications for this question:

(a) The constant c mentioned in the hint refers to the constant c in the definition of O, which presumedly you'll use to prove $T(n) = O(n)$.

(b) If you have no issues with the base case for this question, that's OK and you don't need to use this clarification. Some may be troubled by the fact that not all n will reduce down to precisely 1. If this bothers you, you may assume that $1 \leq T(n) \leq k$ for some fixed constant k for all n such that $1 \leq n < \max(1/a_1, 1/a_2)$. While this is a more precise way of handling this issue, the text misses this issue and there are no notes in the errata page, so apparently it is not considered something we need to worry about. Full credit will be granted whichever way this issue is handled.

Question 9. Remember that a recurrence of the form $T(n) = aT(n/b) + f(n)$ means that the original problem can be split into $\boldsymbol{a}$ subproblems, each of which are $1/b$ times the size of the original problem, and it takes $f(n)$ time (total) to do the splitting and to combine the results of the subproblems into a solution for the original problem.

Question 11. $T(1) = T(2) = 1$. Alternatively, you may use $T(0) = 0, T(1) = 1$. as is done in CLR.