FALL 2007: **COT 5407** INTRO. TO ALGORITHMS

[HOMEWORK 1; DUE SEP 13 AT START OF CLASS]

General submission guidelines and policies: On page 2. Also see separate handout on course web page. ADD A SIGNED STATEMENT THAT YOU HAVE ADHERED TO THE COLLABORATION POLICY FOR THIS CLASS AND THAT WHAT YOU ARE PRESENTING IS YOUR OWN WORK. Without this statement, your homework will not be graded.

Problems

1. (**Regular**) (Exercise 3.1-2, p50) Show that for any real constants a and b, where $b > 0$,
$$(n + a)^b = \Theta(n^b).$$
Note that $f(n) = \Theta(g(n)$ if $f(n) = O(g(n))$ and $g(n) = O(f(n))$.

2. (**Regular**) (Exercise 3-2(e), p58) If $f(n) = n^{\log_2 c}$ and $g(n) = c^{\log_2 n}$, indicate which of these relationships are true and prove your answers: $f(n) = O(g(n))$, $f(n) = \Omega(g(n))$, and $f(n) = \Theta(g(n))$.

3. (**Regular**) (Exercise 3-4(b), p59) Prove or disprove:
$$f(n) + g(n) = \Theta(\min(f(n), g(n))).$$

4. (**Regular**) (Exercise 4-1(c,d), p85) Solve the following recurrences, assuming that $T(n)$ is constant for $n < 2$.
   (a) $T(n) = 16T(n/4) + n^2$.
   (b) $T(n) = 7T(n/3) + n^2$.

5. (**Exercise**) Write down the time complexities of performing LINEARSEARCH and BINARYSEARCH in a sorted array of n elements.

6. (**Exercise**) Write down the precise **invariants** for each of the following algorithms: SELECTIONSORT, INSERTIONSORT, BUBBLESORT, SHAKERSORT, and MERGE (not MERGESORT).

7. (**Extra Credit**) In our first class (Aug 28), we discussed and analyzed a simple algorithm for the SEARCH problem. We discussed two variants – one where X, the number to be searched, was bounded on both sides, and another where x was bounded below, but unbounded above. Binary search was the best strategy for the first version. The best strategy for the second version involved doing a doubling search followed by a binary search. This could be thought of as doing a LINEARSEARCH for m, the smallest exponent of 2 greater than x. What if we consider doing doubling search for m? Can we push this even further? Analyze the best algorithm for this problem.

General submission guidelines: Since people tend to scribble on handwritten homework, you are required to type up your assignment and print it out. Problems are labeled as (**Exercise**), (**Regular**), or (**Extra Credit**). (**Exercise**) are to be turned in, but will not be graded. (**Regular**) problems are to be turned in and will be graded. (**Extra Credit**) problems need not be turned in. They will be graded, but credit will be given only if it is completely correct. (**Extra Credit**) scores will be used **only** if your grade is on the borderline between two grades.

For every algorithmic question, clearly indicate its **Input** and its **Output**. Your pseudocode must contain line numbers (like in the text) and must be properly indented. While pseudocode is generally preferred, you may write formal code using a programming language such as C++ or Java (less desirable). Variable names must be meaningful. If a section of the code is complicated, it must be commented.

Pay careful attention to the final written solution. Reread your written solutions and look for typographical and logical errors. A well-written solution shows clarity of thought and is likely to receive better grades. Not all problems will be graded and not all graded problems will have equal score. If more than one correct algorithm can solve a problem, then a more efficient solution will fetch more credit. Draw pictures whenever possible.

Collaboration Policy (August 30, 2007): Solving an algorithmic problem is a creative process. When presented with a new problem, it is your task to "take it apart" and reach your own understanding. This is a painstaking and time-consuming process. There is much to be learned from the process of thinking out solutions to the assigned homework problems. Getting help from elsewhere destroys this process. However, discussing with others after you have spent some time with a problem can help the process and bring other aspects of the problem to light. You may discuss homework problems with me or with other students in your class, after you have given it sufficient thought. But when the time comes to write up your solution, it MUST be your own work, and it MUST be in your own words. If after working on a problem yourself, you have been unable to solve it satisfactorily, then you may get help from other people, textbooks, or the internet. If you received help from any other source, it is necessary to *cite* your source at the appropriate location in the homework, i.e., write down the URL or the name of the person or the author of the text from which your solution was acquired. GIVE EVEN THE DEVIL IT'S DUE! After getting help from some source, make an attempt to write down the solution in your own words. If you discussed with a classmate or friend and came up with a solution together, then both of you should indicate this in your homeworks. If you are helping someone or providing your solution to someone, make sure they write down you as the source of the solution. You, too, may indicate that you helped this person with the specified problem. If you do not write down where you got help from, it would be considered as cheating. Any evidence of cheating (without citing the source) will result in severe penalization of all parties involved. If this policy gets refined over the course of this semester, this will be posted on the course webpage.