Computer Science 830: Design and Theory of Algorithms

Homework #1
Due September 15th 2011, 10:20am

Each full problem should be solved on a separate sheet of paper to facilitate grading. Limit the solution of each sub-problem to at most one side of a sheet of paper. Please do not wait until the last minute to look at the problems.

1. (Points: 18%) Log onto arctic.cse.msu.edu and cd to the directory ~cse830/Public/hw1/ where you will find a collection of six short C language programs and six unidentified executable files. Each executable accepts an integer value *n* as its command line argument. Using the Unix `time` command, run each executable for multiple values of n (run "`man time`" for more information on how the command works). Remember to choose your values of n carefully in order to get a good idea of what these curves really look like. For example, you might start with: `time exe-A 1`

- Examine the code files. **Find a tight asymptotic-upper bound** (big-O) for each.
- **Graph the run times** for the executables. Make sure the graphs are clear enough to justify your next answer.
- Look at the source code of the C language programs. **Match each executable** with its source program.

2. (Points: 18%) Give a proof or counter-example for each of the following statements:

a. if $f(n) = O(F(n))$ and $g(n) = O(G(n))$, then $f(n)/g(n) = O(F(n)/G(n))$

b. $f(n) = O(g(n))$ implies that $2^{f(n)} = O(2^{g(n)})$

c. For all functions $f(n)$ and $g(n)$, either $f(n) = O(g(n))$ or $g(n) = O(f(n))$.

3. (Points: 18%) For each of the following pairs of functions $f(n)$ and $g(n)$, determine whether $f(n) = O(g(n))$, $g(n) = O(f(n))$, both, or neither.

a. $f(n) = (n^2 - n)/2$, $g(n) = 6n$

b. $f(n) = n + 2\ \text{sqrt}(n)$, $g(n) = n^2$

c. $f(n) = n \log n$, $g(n) = n\ \text{sqrt}(n) / 2$

d. $f(n) = \text{sqrt}(n)$, $g(n) = n + \log n$

e. $f(n) = 2(\log n)^2$, $g(n) = \log n + 1$

f. $f(n) = 4n \log n + n$, $g(n) = (n^2 - n) / 2$

4. (Points: 9%) For each of the following function pairs $f(n)$ and $g(n)$, give a minimal positive constant c such that $f(n) < c\ g(n)$ for all $n \geq 1$.

a. $f(n) = n^2 + n + 1, \quad g(n) = 2n^3$
b. $f(n) = n\ \text{sqrt}(n) + n^2, g(n) = n^2$
c. $f(n) = n^2 - n + 1, \quad g(n) = n^2/2$

5. (Points: 9%) Order of magnitude calculations: These questions will explore your ability to estimate numbers within a couple of orders of magnitude. Don't worry about the *correct* answer, I'm interested in how you make your assumptions and follow through on them.

- Estimate the total amount of money in an armored car completely filled with $20 bills.
- Estimate how many cubic miles of water flow out of the mouth of the Mississippi River each day.
- Estimate the number of Lego bricks it would take to build a copy of the statue of liberty that is the same size as the original.

Do not look up any supplemental facts, but feel free to ask me any questions you have about the exercise. Describe all assumptions you made in arriving at your answer.

6. (Points: 9%) Show that the solution to $T(n) = 2T(\lfloor n/2 \rfloor + 17) + n$ is $O(n \lg n)$.

7. (Points: 10%) Draw the recursion tree for $T(n) = 4T(\lfloor n/2 \rfloor) + cn$, where c is a constant, and pro- vide a tight asymptotic bound on its solution. Verify your bound by the substitution method.