CSE332 Data Abstractions, Summer 2010

Homework 1

Due Friday, July 2, 2010 at the beginning of class.

Problem 1: Recurrence Relations

Consider the following recurrence relation, similar to one seen in lecture 3:

T(1)=5, and for n greater than 1, $T(n) = 1 + 2T(\lfloor n/2 \rfloor)$

Note: $\lfloor n/2 \rfloor$ is the 'floor' of 'n/2': it rounds down to the next largest integer.

a. Give T(n) for n=integers 1 through 8

b. Expand the recurrence relation to get the closed form, as seen in lecture #3. Show your work; do not just the final equation.

Problem 2: Induction

For the following inductive proofs, clearly state the base case, induction step and inductive hypothesis, as described in lecture.

a. Prove Weiss 1.12.a by induction

b. Prove the following by induction: $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

Problem 3: Run-times

Say we have 4 versions of a program we'd like to run on some input n. Each takes a certain amount of time to run, as a function of n:

- P1 takes n days to run
- P2 takes n^2 days to run
- P3 takes 2^n days to run
- P4 takes $\log_2 n$ days to run

So, to run P2 on an n of 4 would take 16 days.

a. For each version of the program, calculate the value of n (rounded down) we could compute if we let the program run for 12 billion years , which is (very) roughly how long until the Earth's sun dies. You can leave the answer in scientific notation, or as 2 raised to some power in scientific notation.

b. Let's say we have access to a computer which runs one million times faster than the one above; so we could compute P1(1) in one millionth of a day. Write out the n we could compute for each of the above algorithms given this new processing power.

Problem 4: Fun with Induction

The following statement is clearly not true. Can you spot the error in the inductive "proof" below? Specify which one of the following 5 numbered lines is wrong, and clearly describe the error.

All cats are the same color "proof": The proof is by induction on n:
Base case (n = 1):

1. If there is only one cat in the set, then the statement trivially holds.

Induction step: (n = k+1). Assume the statement holds for n = k; this is our inductive hypothesis. Now suppose you have k+1 cats.

2. Set the first cat aside. The remaining k must be the same color (let's say black) by our inductive hypothesis.

3. All we have to do now is show that the first one is also black.

4. To do this, remove a second cat and put the first cat back in to form a new set of size k. By the inductive hypothesis, all the cats in the new set are also the same color.

5. Since this set contains k − 1 cats that we already know are black, it follows that they are all black (including the first). This proves the induction step, and so, given the base case, all cats must be the same color.

Problem 5. Big Oh Notation

For each of the following, either prove true (using our definitions for Big Oh, Big Omega and Big Theta, as appropriate) or explain why it is false:

a. If we have an algorithm that runs in O(n) time, and make some changes that cause it to run 10 times slower, it will still run in O(n) time.

b. If $f(n) = O(g(n))$ and $h(n) = O(k(n))$, then $f(n) - h(n) = O(g(n) - k(n))$.

c. If $f(n) = O(g(n))$ and $h(n) = O(k(n))$, then $f(n) + h(n) = O(g(n) + k(n))$.

d. $(2^{n+1}) = \Theta(2^n)$

e. $(2^n)^{1/3} = \Theta(2^n)$

Problem 6. Algorithm analysis

Weiss 2.7a (give the best big-O bound you can for each of the 6 program fragments; you do not need to explain why).