CSE 332 Data Abstractions, Winter 2012

## Homework 1

Due Friday, Jan 13, 2012 at the beginning of lecture. Please be sure your work is readable (either written clearly or typed).

## Problem 1. Run-times

This problem gives an orthogonal view of comparative running times from that given in Figure 2.2 of the textbook. Be sure to look at the patterns in your table when you have completed it.

For each function $f(n)$ and time $t$ in the following table, determine the largest size $n$ of a problem that can be solved in time $t$, assuming that the algorithm to solve the problem takes $f(n)$ microseconds. For large entries (say, those that warrant scientific notation), an estimate is sufficient. For one of the rows, you will not be able to solve it analytically, and will need a calculator or small program.

|  | 1 second | 1 minute | 1 hour | 1 day | 1 month | 1 year |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| $1000 \log _{2} n$ |  |  |  |  |  |  |
| 100 n |  |  |  |  |  |  |
| $100 \mathrm{n} \log _{2} \mathrm{n}$ |  |  |  |  |  |  |
| $10 \mathrm{n}^{2}$ |  |  |  |  |  |  |
| $\mathrm{n}^{3}$ |  |  |  |  |  |  |
| $2^{\mathrm{n} *} 1 / 10$ |  |  |  |  |  |  |

## Problem 2. Recurrence Relations

Consider the following recurrence relation:
$T(1)=5$, and for $n$ greater than $1, T(n)=1+2 T(\lfloor n / 2\rfloor)$
Note: $\lfloor n / 2\rfloor$ is the 'floor' of ' $n / 2$ ': it rounds down to the next largest integer.
a. Give $\mathrm{T}(\mathrm{n})$ for $\mathrm{n}=$ integers 1 through 8.
b. Expand the recurrence relation to get the closed form. Show your work; do not just show the final equation.

## Problem 3. 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 $\mathrm{O}(\mathrm{n})$ 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. $\left(2^{n+1}\right)=\Theta\left(2^{n}\right)$
e. $\left(2^{n}\right)^{1 / 3}=\Theta\left(2^{n}\right)$

## 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"
The proof is by induction on n (the number of cats):
Base case ( $n=1$ ):

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

Induction step: $(\mathrm{n}=\mathrm{k}+1)$. Assume the statement holds for $\mathrm{n}=\mathrm{k}$; 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 $\mathrm{k}-1$ 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. 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). The problem is the same in the second and third editions of Weiss.

