CSE 332: Data Abstractions
Assignment #1
September 24, 2014
due: Wednesday, October 1, 12:30 p.m., *before lecture begins*

Bundles: The problems in each written homework assignment will be divided into 2 groups (to facilitate distribution to grading TAs). You will turn in 2 corresponding bundles. Write your name in the *upper left corner* of each bundle's top page, with your last name printed clearly in CAPITAL LETTERS. Each bundle should be stapled separately. We don't supply the stapler.

This week's turnin bundles: (A) problems 1–2, (B) problems 3–5.

There are 2 pages to this assignment.

1. 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; give those answers in scientific notation, except in the first row give the answers in the form 2^x, where x is in scientific notation. 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
$500 \log_2 n$						
$50n$						
$50n \log_2 n$						
$5n^2$						
$\frac{1}{2} \cdot n^3$						
$\frac{1}{20} \cdot 2^n$						

2. Prove or disprove: $n \log_b n \in \Theta(n \log_a n)$, for any two constants a and b.

3. (a) Prove that $n \ln n \in O(n^{1+\epsilon})$, for any constant real number $\epsilon > 0$. (Hint: choose $c = 1$. I see a way of doing this using derivatives, and there are probably other ways as well. If you have trouble with this, start with the case $\epsilon = 1$.)

   (b) Prove that $n \ln n \notin O(n)$.

4. Let $T(n)$ be the running time of the following procedure on inputs of size n. Find a function $f(n)$ such that $T(n) \in \Theta(f(n))$, and justify your answer.

```
procedure triple(integer n):
   for i from 1 to n do
      for j from −n to 12n do
         for k from j to j + 500 do
            if j − k is even
               then x ← x + 1;
               else x ← 2 * x;
```

5. Let $T(n)$ be the running time of the following procedure on inputs of size n. Find a function $f(n)$ such that $T(n) \in \Theta(f(n))$, and justify your answer.

```
procedure double(integer n):
   for i from 1 to n do
      for j from i + 1 to n do
         x ← x + 1;
```