CSC 383 Sections 401, 410 Fall 2010
Homework Assignment 2

Due: Wednesday, September 29, 1:30 PM

Note: In the problems below, assume throughout that $log(n)$ refers to $log_2(n)$. Also, for problems 2 and 3, you should write an explanation of how you arrived at your answer. This explanation does not need to be a formal "proof". However, without some explanation of how you arrived at your answer, I cannot give partial credit if your answer is incorrect.

Finally, some of the answers to these problems may include additional higher-order polynomials other than the 7 functions listed in the text that characterize the complexities of algorithms.

1. Order the following functions by their Θ-complexity in terms of n.

   $4 \cdot nlog(n) + 2 \cdot n$
   $3 \cdot n + 100 \cdot log(n)$
   $n^2 + 10 \cdot n$
   2^{10}
   $6 + 8 \cdot log(n)$
   $.1 \cdot n^3 + 10 \cdot n^2$
   $2^n + n$

2. Demonstrate the Θ-complexity, in terms of n, of the worst-case running times of the following functions:

   (a) $f(n) = n^2 + 4 \cdot n + 3$
   (b) $f(n) = n^3 - 2 \cdot n^2 - 1$
   (c) $f(n) = 3 \cdot n + log(n)$

   In order to demonstrate that $f(n) = \Theta(g(n))$, you must demonstrate that $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$, where $g(n)$ is one of the following functions:

   1
   $log(n)$
   n
   $n \cdot log(n)$
   n^2
   n^3
   2^n

That is, you must find 2 constants C_1 and C_2, such that $f(n) \leq C_1 \cdot g(n)$ for $n \geq x_1$ (x_1 is a positive integer), and $f(n) \geq C_2 \cdot g(n)$ for $n \geq x_2$ (x_2 is a positive integer). First, determine $g(n)$ for each of the funtions $f(n)$ listed above, such that $f(n) = \Theta(g(n))$. Then find the constants C_1 and C_2 as specified above, and explain why C_1 and C_2 meet the criteria to show that each $f(n) = \Theta(g(n))$.

3. Give a Θ characterization, in terms of n, of the worst-case running time of each of the code fragments below. Assume that n has been declared appropriately and has been set to a value.

   (a)
   ```
   int sum = 0;
   for (int i = 0; i < n; i++ )
     sum++;
   ```

   (b)
   ```
   int sum = 0;
   for (int i = 0; i < n; i++ )
     for (int j = 0; j < n * n; j++ )
       sum++;
   ```

   (c)
   ```
   int sum = 0;
   for ( int i = 0; i < n; i++)
     for ( int j = 0; j < i; j++)
       sum++;
   ```

   (d)
   ```
   int sum = 0;
   for ( int i = 0; i < n; i++)
     for ( int j = 0; j < i * i; j++ )
       for ( int k = 0; k < j; k++ )
         sum++;
   ```