CMPS 101
Midterm 1
Review Problems

Problem 19 in this handout may not be included. It depends on how far we get by Wednesday.

1. Let $f(n)$ and $g(n)$ be asymptotically non-negative functions which are defined on the positive integers.
   a. State the definition of $f(n) = O(g(n))$.
   b. State the definition of $f(n) = \omega(g(n))$

2. State whether the following assertions are true or false. If any statements are false, give a related statement which is true.
   a. $f(n) = O(g(n))$ implies $f(n) = o(g(n))$.
   b. $f(n) = O(g(n))$ if and only if $g(n) = \Omega(f(n))$.
   c. $f(n) = \Theta(g(n))$ if and only if $\lim_{n\to\infty}(f(n)/g(n)) = L$, where $0 < L < \infty$.

3. Prove that $\Theta(f(n)) \cdot \Theta(g(n)) = \Theta(f(n) \cdot g(n))$. In other words, if $h_1(n) = \Theta(f(n))$ and $h_2(n) = \Theta(g(n))$, then $h_1(n) \cdot h_2(n) = \Theta(f(n) \cdot g(n))$.

4. Let $f(n)$ and $g(n)$ be asymptotically positive functions (i.e. $f(n) > 0$ and $g(n) > 0$ for all sufficiently large n), and suppose that $f(n) = \Theta(g(n))$. Does it necessarily follow that $\frac{1}{f(n)} = \Theta\left(\frac{1}{g(n)}\right)$? Either prove this statement, or give a counter-example.

5. Give an example of functions $f(n)$ and $g(n)$ such that $f(n) = o(g(n))$ but $\log(f(n)) \neq o(\log(g(n)))$. (Hint: Consider $n!$ and n^n and use the corollary to Stirling's formula in the handout on common functions.)

6. Let $g(n)$ be an asymptotically non-negative function. Prove that $o((g(n)) \cap \Omega(g(n)) = \varnothing$.

7. Use limits to prove the following (these are some of the exercises at the end of the asymptotic growth rates handout):
   a. If $P(n)$ is a polynomial of degree $k \geq 0$, then $P(n) = \Theta(n^k)$.
   b. For any positive real numbers α and β: $n^\alpha = o(n^\beta)$ iff $\alpha < \beta$, $n^\alpha = \Theta(n^\beta)$ iff $\alpha = \beta$, and $n^\alpha = \omega(n^\beta)$ iff $\alpha > \beta$.
   c. For any positive real numbers a and b: $a^n = o(b^n)$ iff $a < b$, $a^n = \Theta(b^n)$ iff $a = b$, and $a^n = \omega(b^n)$ iff $a > b$.
   d. $f(n) + o(f(n)) = \Theta(f(n))$.

8. Let $g(n) = n$ and $f(n) = n + \frac{1}{2}n^2(\sin(n)+1)$. Show that

   a. $f(n) = \Omega(g(n))$

   b. $f(n) \neq O(g(n))$

   c. $\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right)$ does not exist, even in the sense of being infinite.

   Note: this is the 'Example C' mentioned in the handout on asymptotic growth rates.

9. Use Stirling's formula: $n! = \sqrt{2\pi n}\cdot\left(\frac{n}{e}\right)^n\cdot\left(1+\Theta(1/n)\right)$, to prove that $\log(n!) = \Theta(n\log n)$.

10. Use Stirling's formula to prove that $\binom{2n}{n} = \Theta\left(\frac{4^n}{\sqrt{n}}\right)$.

11. Consider the following *sketch* of an algorithm called ProcessArray which performs some unspecified operation on a subarray $A[p\cdots r]$.

    <u>ProcessArray(*A*, *p*, *r*)</u> (Preconditions: $1 \le p$ and $r \le \text{length}[A]$)
    1. Perform 1 basic operation
    2. if $p < r$
    3. $\quad q \leftarrow \left\lfloor \frac{p+r}{2} \right\rfloor$
    4. $\quad$ ProcessArray(A, p, q)
    5. $\quad$ ProcessArray(A, q+1, r)

    a. Write a recurrence formula for the number $T(n)$ of basic operations performed by this algorithm when called on the full array $A[1\cdots n]$, i.e. by ProcessArray(*A*, 1, *n*). (Hint: recall our analysis of MergeSort.)

    b. Show that the solution to this recurrence is $T(n) = 2n-1$, whence $T(n) = \Theta(n)$.

12. Consider the following algorithm which does nothing but waste time:

    <u>WasteTime(*n*)</u> (pre: $n \ge 1$)
    1. if $n > 1$
    2. $\quad$ for $i \leftarrow 1$ to n^3
    3. $\quad\quad$ waste 2 units of time
    4. $\quad$ for $i \leftarrow 1$ to 7
    5. $\quad\quad$ WasteTime$\left(\lceil n/2 \rceil\right)$
    6. $\quad$ waste 3 units of time

    a. Write a recurrence formula which gives the amount of time $T(n)$ wasted by this algorithm.

    b. Show that when n is an exact power of 2, the solution to this recurrence relation is given by $T(n) = 16n^3 - \frac{1}{2} - \frac{31}{2}n^{\lg 7}$, and hence $T(n) = \Theta(n^3)$.

13. Define $T(n)$ by the recurrence formula

$$T(n) = \begin{cases} 1 & 1 \le n < 3 \\ 2T(\lfloor n/3 \rfloor) + 4n & n \ge 3 \end{cases}$$

Use Induction to show that $\forall\, n \ge 1$: $T(n) \le 12n$, and hence $T(n) = O(n)$.

14. Prove that all trees on n vertices have $n-1$ edges. Do this by (a) induction on the number of vertices, and (b) by induction on the number of edges.

15. Define $S(n)$ for $n \in Z^+$ by the recurrence:

$$S(n) = \begin{cases} 0 & \text{if } n = 1 \\ S(\lceil n/2 \rceil) + 1 & \text{if } n \ge 2 \end{cases}$$

Use induction to prove that $S(n) \ge \lg(n)$ for all $n \ge 1$, and hence $S(n) = \Omega(\lg n)$.

16. Let $f(n)$ be a positive, increasing function that satisfies $f(n/2) = \Theta(f(n))$. Show that

$$\sum_{i=1}^{n} f(i) = \Theta(nf(n))$$

(Hint: Emulate the **Example** on page 4 of the handout on asymptotic growth rates in which it is proved that $\sum_{i=1}^{n} i^k = \Theta(n^{k+1})$ for any positive integer k.)

17. Use the result of the preceding problem to give an alternate proof of $\log(n!) = \Theta(n \log(n))$ that does not use Stirling's formula.

18. Let $T(n)$ be defined by the recurrence formula

$$T(n) = \begin{cases} 1 & n = 1 \\ T(\lfloor n/2 \rfloor) + n^2 & n \ge 2 \end{cases}$$

Show that $\forall n \ge 1$: $T(n) \le \frac{4}{3}n^2$, and hence $T(n) = O(n^2)$.

****

We may not get far enough for this problem. If we do I'll let you know.

19. Define $T(n)$ by the recurrence formula:

$$T(n) = \begin{cases} 7 & 1 \le n < 3 \\ 2T(\lfloor n/3 \rfloor) + 5 & n \ge 3 \end{cases}$$

a. Use the iteration method to determine a solution to the above recurrence.
b. Use the solution you found in part (a) to determine an asymptotic solution.