CMPS 101 Midterm 1 Review Problems

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. 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 $k \ge 0$, 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. $f(n) + o(f(n)) = \Theta(f(n))$.

5. 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 $\log(n!) = \Theta(n\log n)$.

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

7. 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 exact solution to this recurrence is $T(n) = 2n-1$, whence $T(n) = \Theta(n)$.
   c. Use the Master Theorem to show that $T(n) = \Theta(n)$.

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

```
WasteTime(n)   (pre: n ≥ 1)
if n > 1
    for i ← 1 to n^3
        waste 2 units of time
    for i ← 1 to 7
        WasteTime(⌈n/2⌉)
    waste 3 units of time
```

   a. Write a recurrence formula which gives the amount of time $T(n)$ wasted by this algorithm.
   b. Use the Master Theorem to find an asymptotic solution to this recurrence.

9. 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.

10. Use the Master Theorem to find asymptotic solutions for the following recurrences.
   a. $T(n) = 2T(n/4) + \sqrt{n}$
   b. $T(n) = 7T(n/3) + n^2$
   c. $T(n) = 7T(n/3) + n$

11. 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 an exact solution to the above recurrence.
   b. Use the solution you found in part (a) to determine an asympotic solution.
   c. Use the Master Theorem to determine an asymptotic solution.