1. What is the complexity of the following pseudocode algorithms? Explain how you got your answer. (5 points each)

(a)  
\[
\begin{align*}
x &= 0 \\
\text{for } i &= 1 \text{ to } n \\
\quad \text{for } j &= 1 \text{ to } n \\
\quad \quad x &= x + 1 \\
\quad x &= x + 1
\end{align*}
\]

(b)  
\[
\begin{align*}
x &= 0 \\
\text{for } i &= 1 \text{ to } n \\
\quad \text{for } j &= 1 \text{ to } i \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad // \text{ note this is } i, \text{ not } 1 \\
\quad x &= x + 1
\end{align*}
\]

(c)  
\[
\begin{align*}
k &= n \\
\text{while } (k \geq 1) \\
\quad \text{for } i &= 1 \text{ to } k \\
\quad x &= x + 1 \\
\quad k &= k / 3
\end{align*}
\]

(d)  
\[
\begin{align*}
x &= 0 \\
\text{i} &= 2 \\
\text{while } (i < n) \\
\quad i &= i \times i \\
\quad x &= x + 1
\end{align*}
\]
2. Order these functions from smallest to largest. Be sure to indicate when functions are in the same complexity class. (10 points)

- \( n^2 \log n \)
- \( n \)
- \( n \log \log n \)
- \( \log \log n \)
- \( n^3 \)
- \( n^2 \log n^3 \)
- \( n \log n \)
- \( n^2 \)
- \( n \log n^2 \)
- \( \log n \)

3. Determine whether each statement is true or false. Justify your responses. (2 points each)

- (a) if \( f(n) = \Theta(h(n)) \) and \( g(n) = \Theta(h(n)) \), then \( f(n) + g(n) = \Theta(h(n)) \)
- (b) if \( f(n) = \Theta(g(n)) \), then \( c \cdot f(n) = \Theta(c \cdot g(n)) \) for any \( c > 0 \).
- (c) if \( f(n) = O(g(n)) \), then \( g(n) = O(f(n)) \)
- (d) if \( f(n) = \Theta(g(n)) \), then \( g(n) = \Theta(f(n)) \)
- (e) \( f(n) + g(n) = \Theta(h(n)) \), where \( h(n) = \max\{f(n), g(n)\} \)

4. Solve each of the following recurrences using substitution (unrolling). Show each step of your work. You can assume \( T(1)=1 \) in each case. (10 points each)

- (a) \( T(n) = 2T(n-1) + 1 \)
- (b) \( T(n) = 4T\left(\frac{n}{4}\right) + n \)
- (c) \( T(n) = 4T\left(\frac{n}{2}\right) + n^3 \)

5. **CSCI 502 students only**. Show that the function \( \log_2(8n) \) is \( O(\log_2 n) \). Use the log rules in appendix A and the definition of Big-Oh. Formally prove the claim by finding values for \( c \) and \( n_0 \) so that the definition of Big-Oh holds. Show your work. See page 53 in the text for an example. (10 points)