CSCI 433/502 Algorithms
Self Test
Basic Analysis

1. True or False

(a) ______ If \( f(n) = O(g(n)) \) and \( g(n) = O(h(n)) \), then \( f(n) = O(h(n)) \)
(b) ______ \( n^3 + 8n = \Omega(n^2) \)
(c) ______ \( n^3 + 8n = \Theta(n^2) \)
(d) ______ \( n \log n + n^2 \log n = O(n^2) \)
(e) ______ When we analyze algorithms, we usually consider the worst-case time complexity of the algorithm over all possible inputs.
(f) ______ The reason we drop constants and lower order terms when analyzing an algorithm is that, in the limit, as the size of the problem increases, the higher order term will dominate lower-order terms.

2. Indicate the order of the following functions from smallest to largest.

\[ n^2 \quad n \log \log n \quad \log n \quad n \quad 2^n \]

3. Order the functions below from smallest to largest by placing a 1 next to the smallest, a 2 next to the second smallest, and so on.

______ \( 12n + 13n^2 \)
______ \( 5 \log n + 8 \log \log n \)
______ \( 8n + 3 \log n \)
______ \( 3n^2 \log n + 7n \)

4. For each of the following pairs of functions, answer \( \Theta \) if \( f(n) = \Theta(g(n)) \); otherwise, answer \( \Omega \) if \( f(n) = \Omega(g(n)) \) or answer \( O \) if \( f(n) = O(g(n)) \).

(a) \( f(n) = 2^n + n^3 \quad g(n) = n^4 \)
(b) \( f(n) = 6n^2 + 8 \log n \quad g(n) = 5n + 3n^2 + 4 \)
(c) \( f(n) = 8196n^2 + 5 \quad g(n) = 14n^3 + 2n^2 \)
(d) \( f(n) = 3n \quad g(n) = 6n \log n^2 + 13 \)
5. Analyze each of the following code segments, using big-Oh notation.

(a) 
   ```
   for i = 1 to n
       for j = 1 to m
           for k = 1 to m
               x = x + 1
   ```

(b) 
   ```
   i = 1
   while (i < n) {
       x = x + 1
       i = i * 2
   }
   ```

(c) 
   ```
   for i = 1 to n
       for j = 1 to n
           x = x + 1
       for k = 1 to m
           x = x + 1
   ```

(d) 
   ```
   for i = 1 to n
       for j = 1 to m
           x = x + 1
       for k = 1 to m * m
           x = x - 1
   ```

(e) 
   ```
   while (m >= 1) {
       for i = 1 to n
           x = x + 1
       m = m / 2
   }
   ```

6. Simple complexity:

(a) Name an algorithm to search an unordered array? What is the worst case time complexity of the algorithm?

(b) Name an algorithm to search an ordered array? What is the worst case time complexity of the algorithm?

(c) Name an algorithm to search an unordered linked list? What is the worst case time complexity of the algorithm?

(d) Name an algorithm to search an ordered linked list? What is the worst case time complexity of the algorithm?