1. (30 pts.) Indicate whether true or false. Beware of guessing:

   correct answer +5pts.  incorrect answer −3pts.  no answer 0pts

(a) ___ \((\ln n)^2 \in \Theta(n)\).
(b) ___ \(n \ln n \in o(n^{1.2})\).
(c) ___ \(n^{1.2} \in o(n \ln n)\).
(d) ___ If \(f(n) \in \Theta(g(n))\), then \(g(n) \in \Theta(f(n))\).
(e) ___ If \(f_1(n) \in O(g(n))\) and \(f_2(n) \in O(g(n))\), then \((f_1(n) + f_2(n)) \in O(g(n))\).
(f) ___ Let \(W_M(n)\) and \(W_Q(n)\) be the worst case times for mergesort and quicksort, respectively. True or false: \(W_M(n) \in o(W_Q(n))\).

2. (25 pts.) Consider the following eight complexity categories (remember \(\lg = \log_2\)):

   \(\Theta(2^{\ln n})\)  \(\Theta(2^{\lg n})\)  \(\Theta(n \lg(\lg n))\)  \(\Theta(n \lg n)\)  \(\Theta(n(1+\lg n))\)  \(\Theta(n!)\)  \(\Theta(2^n)\)  \(\Theta(n)\)

(a) Which are equal? (There may be more than one pair.) Give a reason for any equalities.

(b) Arrange the distinct categories in order from slowest growing to fastest growing. In other words, if \(\Theta(f(n))\) is to the left of \(\Theta(g(n))\), then \(f(n) \in o(g(n))\).

3. (20 pts.) It is known that \(T(1) = 0\) and that \(T(n + 1) = 7T(n) + 12\) for \(n > 0\). Prove that \(T(n) = 2(7^{n-1} - 1)\).
4. (25 pts.) In the following algorithm, \(\ldots\) stands for some simple calculations that take constant time.

```
procedure(n)
    for k from 1 to n do
        \(\ldots\) /* produces a number j */
        if k divides j, then mergesort an n-long list
        \(\ldots\)
    end for loop
end
```

*Note:* Think of \(j\) as a random integer, so the probability that “\(k\) divides \(j\)” is \(1/k\).

(a) Suppose the sorting were free (which it is not). What is the complexity class for the average running time of this algorithm. **You MUST give a reason for your answer.** (The class should be of the form \(\Theta(f(n))\) where \(f(n)\) is a simple function.)

(b) Suppose that the basic operation is a comparison in mergesort. What is the complexity class for the average running time of this algorithm. (You may give your answer in the form \(\Theta(\sum f(k))\) where \(f(k)\) is a simple function and the sum runs from 1 to \(n\).) **You MUST give a reason for your answer.**

(c) Use (a) and (b) to find the complexity class for the average running time of this algorithm. **You MUST give a reason for your answer.**