(1) [20 points]

Although merge sort runs in $\Theta(n \lg n)$ worst-case time and insertion sort runs in $\Theta(n^2)$ worst-case time, the constant factors in insertion sort can make it faster in practice for small problem sizes on many machines. Thus, it makes sense to *coarsen* the leaves of the recursion by using insertion sort within merge sort when subproblems become sufficiently small. Consider a modification to merge sort in which $n/k$ sublists of length $k$ are sorted using insertion sort and then merged using the standard merging mechanism, where $k$ is a value to be determined.

a. Show that insertion sort can sort the $n/k$ sublists, each of length $k$, in $\Theta(nk)$ worst-case time.

b. Show how to merge the sublists in $\Theta(n \lg(n/k))$ worst-case time.

c. Given that the modified algorithm runs in $\Theta(nk + n \lg(n/k))$ worst-case time, what is the largest value of $k$ as a function of $n$ for which the modified algorithm has the same running time as standard merge sort, in terms of $\Theta$-notation?

d. How should we choose $k$ in practice?

(2) [5 points]

Express the function $n^3/1000 - 100n^2 - 100n + 3$ in terms of $\Theta$-notation
(3) [10 points]

Let \( f(n) \) and \( g(n) \) be asymptotically nonnegative functions. Using the basic definition of \( \Theta \)-notation, prove that \( \max(f(n), g(n)) = \Theta(f(n) + g(n)) \).

(4) [10 points]

Use a recursion tree to determine a good asymptotic upper bound on the recurrence \( T(n) = 4T(n/2 + 2) + n \). Use the substitution method to verify your answer.

(5) [10 points]

Use a recursion tree to determine a good asymptotic upper bound on the recurrence \( T(n) = 2T(n - 1) + 1 \). Use the substitution method to verify your answer.

(6) [15 points]

Can the master method be applied to the recurrence \( T(n) = 4T(n/2) + n^2 \lg n \)? Why or why not? Give an asymptotic upper bound for this recurrence. Please prove the bound. (Clue: try \( \Theta(n^2 \lg^2 n) \))

(7) [30 points]

Give asymptotic upper and lower bounds for \( T(n) \) in each of the following recurrences. Assume that \( T(n) \) is constant for \( n \leq 2 \). Make your bounds as tight as possible, and justify your answers.

a. \( T(n) = 2T(n/2) + n^4 \).

b. \( T(n) = T(7n/10) + n \).

c. \( T(n) = 16T(n/4) + n^2 \).

d. \( T(n) = 7T(n/3) + n^2 \).

e. \( T(n) = 7T(n/2) + n^2 \).

f. \( T(n) = 2T(n/4) + \sqrt{n} \).