Problem 1: Assume that $k, \epsilon$ are constants with $k \geq 1$ and $0 < \epsilon < 1$. Also log denotes the logarithm in base 2, and ln the natural logarithm in base $e$. State which among $A = O(B)$, $A = \Omega(B)$ and $A = \Theta(B)$ is correct for each pair of function below. Justify your answer. 2 points per question.

- $A = \log^k n$ and $B = n^\epsilon$
- $A = n^k$ and $B = n^{\frac{\ln n}{\ln 3}}$
- $A = 3^n$ and $B = 5^n$
- $A = n^{\ln n}$ and $B = e^{\ln^2 n}$
- $A = 4^n$ and $B = 3^{n^2}$

Problem 2: For each of the following recurrences: (i) describe what kind of "divide and conquer" algorithm would give rise to such a recurrence; (ii) give asymptotic upper and lower bounds on $T(n)$. Make your bounds as tight as possible and justify your answer. Assume $T(n)$ is a constant for $n \leq 2$.

- $T(n) = 4T(n/4) + n \log n$ [2 points]
- $T(n) = \sqrt{n}T(\sqrt{n}) + n$ [4 points]
- $T(n) = 3T(n/3) + \frac{n}{\log n}$ [4 points]

*Hint: What is the depth of the recursion tree when at each step the size of the input goes from $n$ to $\sqrt{n}$. Remember that $\sqrt{n} = n^{1/2}$. So at level $i$ of the recursion tree the input size of each subproblem is $n^{2^{-i}}$. Therefore the depth of the recursion tree is $k$ such that $n^{2^{-k}} = 1$. What is this $k$? Would it help to write $n = 2^m$ where $m = \log n$?*

Problem 3: **BubbleSort** is a sorting algorithm that scans the vector from left to right and swaps to adjacent elements that are out of order. It loops the above scan until all elements are sorted. Write some pseudocode for **BubbleSort**. What is the worst-case running time of **BubbleSort**? Justify your answer.

*Note: The description of **BubbleSort** in the problem above is different than the one in Problem 2.2 in the book, though the ideas are the same. The pseudocode in Problem 2.2 does not match the description above, and therefore you will not get credit if you give that pseudocode as your solution.*