1. Compare the following pairs of functions, and indicate whether \( f = o(g), g = o(f), f = \Theta(g), \) or NOTA (none of the above).

\[
\begin{array}{ll}
f(n) & g(n) \\
\hline
\text{a)} & 100n + \log_2 n \quad (\log_2 n)^2 \\
\text{b)} & \log_2 n \quad \log_3 n \\
\text{c)} & 2^n \quad 3^n \\
\text{d)} & n^2 / \log_2 n \quad n(\log_2 n)^2 \\
\text{e)} & \sqrt{n} \quad (\log_2 n)^5 \\
\text{f)} & n^5 \quad (3n + 10)^5 \\
\text{g)} & 2^n \quad n^{10} \\
\text{h)} & \log_2(n!) \quad \log(n^n)
\end{array}
\]

2. Find a counter-example to the following claim: \( f = O(r) \) and \( g = O(s) \) implies \( f/g = O(r/s) \).

3. Use the “master theorem” to solve the following recurrences:

\[
\begin{align*}
(\text{a)} & \quad T(n) = 4T(n/2) + cn^2 \\
(\text{b)} & \quad T(n) = 5T(n/2) + cn^2 \\
(\text{c)} & \quad T(n) = 3T(n/2) + cn^2
\end{align*}
\]

4. An recursive algorithm \( A \) works on inputs of size \( n \) by solving two subproblems of size at most \( \lceil n/2 \rceil \), and combining the results of these two solutions in time \( O(n \log n) \). Give an asymptotic (big-O) bound for the running time of \( A \). Prove your result using a recursion tree argument.

5. The following recurrence relation appears in divide-and-conquer algorithms in which the problem is divided into unequal size parts:

\[
T(n) \leq \sum_{i=1}^{k} a_i T([n/b_i]) + cn.
\]

Here, the \( a_i \)'s and \( b_i \)'s, as well as \( c \), are positive integer constants. You may assume the above recursively defined bound for \( T(n) \) holds for \( n \geq \max_i b_i \), and that \( T(n) \) is bounded by a constant \( t \) for \( n < \max_i b_i \).

The goal of this exercise is to prove that \( T(n) = O(n) \), assuming \( \delta := \sum_{i=1}^{k} a_i/b_i < 1 \). To this end, instead of using a recursion tree analysis, try using a direct proof by induction. To do this, introduce a constant \( d \) (to be determined by your analysis), and prove by (strong) induction that \( T(n) \leq dn^2 \) for all positive integers \( n \). You should be able to identify exactly where in the proof you use the assumption that \( \delta < 1 \).

6. You are given \( a_1, \ldots, a_n \), where each \( a_i \) is an \( \ell \)-bit integer. Here, \( n \) and \( \ell \) are independent parameters. Using Karatsuba’s algorithm as a subroutine, show how to compute the product \( \prod_{i=1}^{n} a_i \) in time \( O((n\ell)^{\log_2 3}) \).