Problem 1. (2 points) For the following recurrences, obtain a solution using the master method (or explain why not applicable), then give a substitution proof for the upper bound:

(a) \( T(n) = 4T\left(\frac{n}{2}\right) + n \). (CLRS 4.3-7)

(b) \( T(n) = 4T\left(\frac{n}{2}\right) + n^2 \). (CLRS 4.3-8)

(c) \( T(n) = 4T\left(\frac{n}{2}\right) + n^2 \log n \). (CLRS 4.5-4)

Problem 2. (2 points) Find the exact solution of the recurrence:

\[
T(n) = \begin{cases} 
c & \text{if } n = 0, 
aT(n-1) + k & \text{if } n > 0.
\end{cases}
\]

Problem 3. (2 points) Give pseudocode for iterative binary search. Prove correctness of your algorithm using a loop invariant, state the worst-case running time.

Problem 4. (2 points) Give pseudocode for recursive binary search. Formulate and solve a recurrence describing the worst-case running time.

Problem 5. (3 points) Let \( A = (a_1, a_2, ..., a_n) \) be a sequence of \( n \) distinct numbers. A pair \((i, j)\) is called an inversion of \( A \) if \( i < j \) and \( a_i > a_j \). Give pseudocode for a modification of merge sort to count the number of inversions of the sequence. Formulate and solve a recurrence describing the worst-case running time.