Constructing Proofs

Problem 1: Feedback (10 points)
Share a joke or a poem to entertain the class! Can you come up with a joke or a poem that includes the words "induction", "recursion", "order of growth" and/or "algorithm"?

Problem 2: Counting (20 points)
How many 7 English-letter strings are there when
(a). There are exactly three a’s.
(b). There are at least three a’s.
(c). There are exactly two a’s and two b’s.
(d). There are no a’s, no b’s, and there is exactly one c.

Problem 3: (20 points)
Write the definition of the O() notation. Prove that:
(a) \( n + \sqrt{n} + 5 = O(n) \),
(b) \( n^2 + n + \log n = O(n^2) \).

Problem 4: (20 points)
Recall the definition of the log function: \( x^y = y \) if and only if \( \log_y x = y \).
Recall also that \( a^{\log_a x} = x \).
(a) Prove that \( \log_a b = \frac{\log_b}{\log_c a} \).
(b) Using the definition of the O() notation, prove that \( n \log_2 n = O(n \log_e n) \) and that \( \log_2^2 n = O(\log_2^2 n) \).

Problem 5: (10 points)
Argue that:
(a) \( \sqrt{2^{\log_2 n}} = n^{\frac{1}{2}} \).
(b) \( 4^{\log_2 n} = n^2 \).
(c) \( 2^{\log_2^2 n} = n^{\log_2 n} \).

Problem 6: (20 points)
Rank the following functions by order of growth, that is, give an arrangement \( g_1, g_2, \ldots \), such that \( g_1 = O(g_2), g_2 = O(g_3) \), and so on. You do not need to give proofs.
\( \sqrt{2^{\log_2 n}}, n^3, n \log_2 n, \log_2^2 n, 2^{\log_2 n}, 2^{\log_2^2 n}, 4^{\log_2 n} \).