Problem 1. (30%) Define \( f(n) = 1 + c + c^2 + c^3 + \ldots + c^n \) where \( c \) is a positive real number. Prove:

1. \( f(n) = O(n) \) if \( c = 1 \);
2. \( f(n) = O(c^n) \) if \( c > 1 \);
3. \( f(n) = O(1) \) if \( c < 1 \).

Answer.
1. Trivial and omitted.
2. \( f(n) = \frac{c^{n+1} - 1}{c - 1} \). It is easy to verify that \( f(n) \leq c^n \) for all \( n \geq 1 \).
3. \( f(n) = \frac{1 - c^{n+1}}{1 - c} \leq \frac{1}{1 - c} = O(1) \).

Problem 2. (30%) Suppose that you are given \( n \) distinct integers in an array \( A \). All the integers are (i) in the range \([1, 10n^2]\) and (ii) multiples of \( n \). Describe an algorithm to sort \( A \) in \( O(n) \) time.

Answer. First, decrease \( A[i] \) by \( n \) for each \( i \in [1, n] \). This takes \( O(n) \) time. After this, all the integers \( A \) are in the range \([1, 10n]\). Then, perform counting sort on \( A \) in \( O(U + n) = O(10n + n) = O(n) \) time, where \( U \) is the size of the range (which is \( 10n \)). Finally, increase \( A[i] \) by a factor of \( n \) for each \( i \in [1, n] \) in \( O(n) \) time. The array \( A \) at this time is the sorted order.