Problem 1. Let \( f(n) \) be a function of positive integer \( n \). We know:
\[
\begin{align*}
  f(1) &= 1 \\
  f(2) &= 2 \\
  f(n) &= 3 + f(n-2).
\end{align*}
\]
Prove \( f(n) = O(n) \).

Problem 2. Let \( f(n) \) be a function of positive integer \( n \). We know:
\[
\begin{align*}
  f(1) &= 1 \\
  f(2) &= 2 \\
  f(n) &= n/10 + f(n-2).
\end{align*}
\]
Prove \( f(n) = O(n^2) \).

Problem 3. Let \( f(n) \) be a function of positive integer \( n \). We know:
\[
\begin{align*}
  f(1) &= f(2) = \ldots = f(1000) = 1 \\
  \text{and for } n > 1000 \quad f(n) &= 5n + f([n/1.01]).
\end{align*}
\]
Prove \( f(n) = O(n) \). Recall that \([x]\) is the ceiling operator that returns the smallest integer at least \( x \).

Problem 4. Let \( f(n) \) be a function of positive integer \( n \). We know:
\[
\begin{align*}
  f(1) &= 1 \\
  f(n) &= 10 + 2 \cdot f([n/8]).
\end{align*}
\]
Prove \( f(n) = O(n^{1/3}) \).

Problem 5. Let \( f(n) \) be a function of positive integer \( n \). We know:
\[
\begin{align*}
  f(1) &= 1 \\
  f(n) &= f([n/4]) + f([n/2]) + n.
\end{align*}
\]
Prove \( f(n) = O(n) \).

Problem 6. Consider a set \( S \) of \( n \) integers that are stored in an array (not necessarily sorted). Let \( e \) and \( e' \) be two integers in \( S \) such that \( e \) is positioned before \( e' \). We call the pair \((e, e')\) an inversion in \( S \) if \( e > e' \). Write an algorithm to report all the inversions in \( S \). Your algorithm must terminate in \( O(n^2) \) time.

For example, if the array stores the sequence \((10, 15, 7, 12)\), then your algorithm should return \((10, 7), (15, 7), \) and \((15, 12)\).