Big-Oh notation

1. Decide in all possible cases whether \( f_i(n) = O(f_j(n)) \) is true or not if

\[
\begin{align*}
  f_1(n) &= 11n^2, \\
  f_2(n) &= 8n^2 \log n, \\
  f_3(n) &= n^2 + 100000.
\end{align*}
\]

2. (a) Let’s suppose that \( f(n) = O(n^2) \) and \( g(n) = \Theta(n^3) \). Is it true that \( f(n) = O(g(n)) \)?
   (b) Let’s suppose that \( f(n) = O(n^3) \) and \( g(n) = \Theta(n^2) \). Is it true that \( g(n) = O(f(n)) \)?
   (c) Let’s suppose that \( f(n) = O(n^3) \) and \( g(n) = O(n^2) \). Is it true that \( g(n) = O(f(n)) \)? Is it possible that \( f(n) = O(g(n)) \)?

3. Let’s suppose that \( f(n) \) and \( g(n) \) are functions with non-negative values. Prove that

\[
\max(f(n), g(n)) = \Theta(f(n) + g(n))
\]

4. Give a linear algorithm (ie. whose running time is \( O(n) \)) using only comparisons to find the maximum among \( n \) different numbers. What is the precise number of comparisons we have to perform to find the maximum?