Problem 1. Explain what it means for \( f \) to be in \( O(g) \) in your own words.

Problem 2. Order the below functions so that if \( f \) appears before \( g \), then \( f \in O(g) \).

\[
\begin{array}{c}
n! \\
\log n \\
\log n! \\
\log^* n \\
n^3 \\
0 \\
\log 2n \\
n^2 \\
\log n \\
1 \\
2^n \\
3^n \\
n \\
n^{1/2}
\end{array}
\]

Unless otherwise specified, you may assume that \( \log = \log_2 \). Also, \( \log^* n \) is the \textit{iterated logarithm}. It is the number of times you must take the logarithm to get a number between 0 and 1. For example,

\[
\begin{align*}
\log^* 1 &= 0 \\
\log^* 2 &= 1 \\
\log^* 4 &= 2 \\
\log^* 16 &= 3 \\
\log^* 65536 &= 4 \\
\log^* 2^{65536} &= 5
\end{align*}
\]

Problem 3. Prove that \( 3n^2 + 2n - 77 \in O(n^2) \).

Problem 4. Prove that if \( f_1 \in O(g) \) and \( f_2 \in O(g) \), then \( f_1 + f_2 \in O(g) \).