3.1-3
Explain why the statement, “The running time of algorithm A is at least \( O(n^2) \),” is meaningless.

**Theorem 3.1**
For any two functions \( f(n) \) and \( g(n) \), we have \( f(n) = \Theta(g(n)) \) if and only if \( f(n) = O(g(n)) \) and \( f(n) = \Omega(g(n)) \).

3.1-5
Prove Theorem 3.1.

3.1-7
Prove that \( o(g(n)) \cap \omega(g(n)) \) is the empty set.

\[
\begin{align*}
n! &= o(n^n), \\
n! &= \omega(2^n), \\
\lg(n!) &= \Theta(n \lg n)
\end{align*}
\]  \hspace{1cm} (3.19)

3.2-3
Prove equation (3.19). Also prove that \( n! = \omega(2^n) \) and \( n! = o(n^n) \).

3.2-8
Show that \( k \ln k = \Theta(n) \) implies \( k = \Theta(n / \ln n) \).