1. Problem 3-2 (all parts) in the book (page 61). Fix \( k = 1, \epsilon = 1, \) and \( c = 2 \). For each “yes,” find a set of constants (from the definitions, not the problem) such that functions \( A \) and \( B \) satisfy the definition.


3. Consider the following standard, two-argument version of the Ackermann function

\[
A(m, n) = \begin{cases} 
  n + 1 & \text{if } m = 0 \\
  A(m-1, 1) & \text{if } m > 0 \text{ and } n = 0 \\
  A(m-1, A(m, n-1)) & \text{if } m > 0 \text{ and } n > 0 
\end{cases}
\]

where \( m \) and \( n \) are nonnegative integers.

   a. Compute \( A(2, 4) \) by hand. Show your work.
   b. Write code to compute \( A(3, 4) \).
   c. Explain why you don’t want to compute \( A(4, 4) \) using your code.

4. **Programming Assignment:** Consider square matrices, each with \( n \) rows and \( n \) columns. Write a simple implementation of the standard \( O(n^3) \) matrix multiplication algorithm. Compare it to a Strassen’s algorithm implementation of your choice (possible options will be posted on my webpage). Experimentally determine the crossover point, that is, the value of \( n \) at which the Strassen’s implementation is faster than the algorithm you implemented. For a fair comparison, write your code in the same language as the Strassen’s algorithm implementation that you choose.