1. Using the definition of big-O on page 53 of OODS\(^1\), for each of the following functions, if an algorithm takes that many primitive operations on an input of size \( n \), what is the time complexity of the algorithm? **Show that your answer is correct** by choosing suitable values for the constants \( c \) and \( n_0 \). You do not need to give a formal mathematical proof (which is beyond the scope of this course); instead, you may obtain the values of these constants by testing or by informal reasoning. **Explain** how you got your answer.

   (a) \( 6n^2 + 9n + 12 \)
   (b) \( 0.015n\log_2 n + 22n + 126 \log n + 329 \)
   (c) \( 17 \log^2 n + 8 \) \( \text{Note that } \log^2 n \text{ is the same as } (\log n)^2. \)

2. In Figure 1.1 on page 55 of OODS, the value of \( \log 10 \) is 3.322, which is actually \( \log_2 10 \) (log to the base 2 of 10). **Does it matter** whether we use \( \log_2 \), \( \log_{10} \) or \( \log_a \) for some real number \( a > 1 \)? **Explain.** (Hint: what is the relationship between \( \log_p n \) and \( \log_q n \)?)

3. For each of the following problems, **is it possible to write a procedure** that will solve the problem and **that meets the definition of an algorithm** (OODS, page 47)? You don’t need to write the algorithm, but you must **explain your answer**.

   (a) Determine whether two strings are the reverse of each other (i.e., the last character of one is the first of the other, etc).
   (b) Determine whether a given integer \( x \) is a prime number (only factors are 1 and itself).
   (c) Determine whether the countries on a map can be colored with only four colors, so that no two adjacent countries have the same color.
   (d) Consider this formula: \( x^n + y^n = z^n \)
      \( \text{Is } n = 2 \text{ the largest integer value for } n \text{ for which there exist positive integers } x, y \text{ and } z \text{ such that the formula has at least one solution?} \)

4. Write recursive functions in C++ to solve the following recurrence relations.

   (a) Ackerman’s function:
   \[
   A(m, n) = \begin{cases} 
   n + 1, & \text{if } m = 0 \\
   A(m - 1, 1), & \text{if } n = 0 \\
   A(m - 1, A(m, n - 1)), & \text{otherwise}
   \end{cases}
   \]

   (b) Greatest Common Divisor:
   \[
   \text{GCD}(m, n) = \begin{cases} 
   m, & \text{if } n = 0 \\
   n, & \text{if } m \mod n = 0 \\
   \text{GCD}(n, m \mod n), & \text{otherwise}
   \end{cases}
   \]

5. Draw the call tree (OODS, page 62) for the recursive algorithms from the previous question:

   (a) Ackerman’s function for \( m = 5 \) and \( n = 3 \)
   (b) Greatest Common Divisor for \( m = 30 \) and \( n = 75 \)

\(^1\)S. Radakrishnan, L. Wise & C. N. Sekharan, *Object-Oriented Data Structures Featuring C++*, 1999
6. Write a constructor for ArrayClass with the signature

    ArrayClass (int arrayLength, int listLength, Object* list)

that constructs an instance of ArrayClass with arrayLength elements, such that:

- if arrayLength ≥ listLength, then all of the elements of list are copied into
  the first listLength elements of the ArrayClass instance;
- otherwise, the first arrayLength elements of list are copied into the
  ArrayClass instance.

7. Write a function that takes an instance of ArrayClass and returns another instance of
   ArrayClass that has the same elements, except that repeats of the same value are elim-
   inated; for example, applying the function to an array containing 4,12,64,12,92,12,53,92,4
   will return an array containing 4,12,64,92,53. (Hint: use a bucket array.)