1. Once an algorithm has been implemented correctly in C/C++ code, compiled and linked, it can be expected to run and work correctly – true or false?

2. If we are primarily concerned with speed of execution and if the programmer has a choice, which would be the better choice for the main data structure: an array or a linked list? Why?

3. You are to write a program in which the user enters a list of double-precision real numbers. Which of the following can be found without using an array? Explain why or why not in each case:
   3a. The largest number.
   3b. The median of the numbers.
   3c. The average of the numbers.
   3d. The closest two numbers.
   3e. The variance of the numbers.

4. Explain the definitions of $\Theta$ and $O$. Give an example of two functions $f(n)$ and $g(n)$ such that $f(n) \in O(g(n))$ but $f(n) \notin \Theta(g(n))$.

5. Our general framework for analyzing the mathematical complexity of an algorithm requires:
   i. defining allowable data sets with an associated definition of size $n$.
   ii. specifying an algorithm which will operate on the data sets with a complexity $C(n)$ depending on one or more basic (or critical) operations $c_{op}$.
   iii. obtaining the result(s).
   iv. specifying a second algorithm to verify the correctness of the result(s) — this second algorithm also has a complexity.

Since the efficiency of the algorithm may depend on the data set, we could get different complexities: $C_{ave}(n)$ and $C_{worst}(n)$. What is the difference between the two? Are there algorithms whose complexity doesn’t depend on the data set?

6. For each of the following functions classify their complexity as one of: constant, linear, quadratic, cubic, $\Theta(n \log n)$, polynomial of degree greater than three, exponential, worse than exponential, or none of the previous.
   i. $f(n) = \sqrt{n^4 + 34n^3 + 7n - 3}$.
   ii. $g(n) = (7n - 3) \log_{10} n^4$.
   iii. $h(n) = \sum_{k=1}^{n} 2^k$.
   iv. $j(n) = \sum_{k=1}^{n} 5k$.
   v. $r(n) = 2^{\log_2 n} + \sqrt{n^2 + 1 \log_2 n^2}$.
   vi. $s(n) = n^n$.
   vii. $t(n) = (\log_{10} n)^3$.

7. In addition to mathematical complexity, we can implement an algorithm in C/C++ and test how long it takes to process a data set. Three different times can be recorded: real, user (or virtual), and system. Explain the differences between these.

8. Let the size of an unsigned integer be $n = \text{the number of bits needed for its binary representation}$. What is the complexity $C(n)$ of finding its smallest prime factor if we use the sieve of Erastosthenes?
9. Shannon entropy can be used to measure qualitatively how far a list is from being sorted (in increasing order). Is it true to say that “the smaller the entropy, the better the sorting algorithm will perform?”

10. If the basic operation is multiplication, find an efficient algorithm for evaluating a polynomial:

\[ p(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0 \]

What is the resulting complexity as a function of the degree \( n \)?

11. Consider the following block of code (the full program is not shown) in which \{a[]} is an array of num double precision reals, and swap() is a procedure which swaps two doubles (passed by reference). What does the code output when it is executed? What is the complexity?

```c
int i;
for (i = 1; i < num; i++) {
    if (a[i] < a[0]) swap(a[0], a[i]);
    if ((i >= 2) && (a[i] < a[1])) swap(a[1], a[i]);
}
cout << a[1];
```

12. A certain algorithm computes something about \( n \) points \( \{p_1, p_2, p_3, \ldots, p_n\} \). It requires that a certain computation be performed on each distinct pair of points (without regard to order). How many operations will be performed? What is the complexity of the algorithm?

13. A certain algorithm computes something about \( n \) vertices of a graph \( \{v_1, v_2, v_3, \ldots, v_n\} \). It requires that for each ordered triple \( (v_{i_1}, v_{i_2}, v_{i_3}) \), (where order matters but duplication is not allowed), a certain calculation be performed. How many operations will be performed? What is the complexity of the algorithm?

14. Consider the two blocks of code below. For the purposes of this question ignore the problem of overflow. It is desired to compute power = a\(^n\) where a is a constant and \( n \) is the size of the problem. If the basic operation is multiplication, find the complexity for each block of code:

```c
int i;
double power=1.0;
for (i = 1; i <= n; i++)
    a = power*a;

double power=1.0;
double twos=a;
while (n > 0) {
    if (n & 0x00000001) {
        power = power*twos;
    }
    twos = twos * twos;
    n = n >> 1;
}
```

15. Suppose that a function \( M \) has initial value \( M(1) = 1 \) and recursion equation

\[ M(n) = 2M(n-1) + 1 \]

Find the value of \( M(n) \) as a formula.

16. It is often stated that insertion sort, in general, makes fewer comparisons but more moves than selection sort. Do you agree with this? Why or why not?