**Question 1** [Section 3.1 Exercise 12, p.202 (10 points)]

Describe an algorithm that uses only assignment statements that replaces the triple \((x, y, z)\) with \((y, z, x)\). What is the minimum number of assignment statements needed? Express your algorithm in pseudocode.

**Question 2** [Section 3.1 Exercise 32, p.203 (10 points)]

Devise an algorithm that finds all terms of a finite sequence of integers that are greater than the sum of all previous terms of the sequence. Express your algorithm in pseudocode.

**Question 3** [Section 3.1 Exercise 34, 38, p.203 (10 points)]

**a)** Use the bubble sort to sort \(\{6, 2, 3, 1, 5, 4\}\), showing the lists obtained at each step.

**b)** Use the insertion sort to sort the same list, showing the lists obtained at each step.

**Question 4** [Section 3.2 Exercise 22, p.216 (10 points)]

Arrange the functions \((1.5)^n, n^{100}, (\log n)^3, \sqrt{n} \log n, 10^n, (n!)^2\) and \(n^{99} + n^{98}\) in a list so that each function is \(O\) of the next function.

**Question 5** [Section 3.2 Exercise 26, p.217 (10 points)]

Give a \(O\) estimate for each of these functions. For the function \(g\) in your estimate \(f(x)\) is \(O(g(x))\), use a simple function \(g\) of smallest order.

**a)** \((n^3 + n^2 \log n)(\log n + 1) + (17 \log n + 19)(n^3 + 2)\)

**b)** \((2^n + n^2)(n^3 + 3^n)\)

**c)** \((n^n + n^2 + 5^n)(n! + 5^n)\)

**Question 6** [Section 3.2 Exercise 44, p.217 (10 points)]

Suppose that \(f(x), g(x),\) and \(h(x)\) are functions such that \(f(x)\) is \(\Theta(g(x))\), and \(g(x)\) is \(\Theta(h(x))\). Show that \(f(x)\) is \(\Theta(h(x))\).

**Question 7** [Section 3.3 Exercise 14, p.230 (10 points)]

(Horner’s method) The following pseudocode shows how to use Horner’s method to find the value of \(a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0\) at \(x = c\).

**procedure** Horner \((c, a_0, a_1, \ldots, a_n : \text{real numbers})\)

\[
\begin{align*}
y &:= a_n \\
\text{for } i &:= 1 \text{ to } n \\
y &:= y \ast c + a_{n-i} \\
\text{return } y \{y = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0\}
\end{align*}
\]

**a)** Evaluate \(3x^2 + x + 1\) at \(x = 2\) by working through each step of the algorithm showing the values assigned at each assignment step.
b) Exactly how many multiplications and additions are used by this algorithm to evaluate a polynomial of degree \( n \) at \( x = c \)? (Do not count additions used to increment the loop variable.)

**Question 8** [Section 3.3 Exercise 39, p.231 (10 points)]

Describe how the number of comparisons used in the worst case changes when these algorithms are used to search for an element of a list when the size of the list doubles from \( n \) to \( 2n \), where \( n \) is a positive integer.

a) linear search

b) binary search

**Question 9** [Section 3.3 Exercise 44, p.231 (10 points)]

Assume that the number of multiplications of entries used to multiply a \( p \times q \) matrix and a \( q \times r \) matrix is \( pqr \). What is the best order to form the product \( ABC \) if \( A, B \) and \( C \) are matrices with dimensions \( 3 \times 9, 9 \times 4 \) and \( 4 \times 2 \), respectively?

**Question 10** ([10 points])

a) On what kind of data will bubble sort algorithm work in its (i) worst case complexity; (ii) best case complexity, respectively?

b) On what kind of data will insertion sort algorithm work in its (i) worst case complexity; (ii) best case complexity, respectively?

**Question Bonus Question** [1 point for course total]

Let \( f(n) \) and \( g(n) \) be asymptotically positive functions. Prove or disprove each of the following conjectures.

a) \( f(n) \) is \( O(g(n)) \) implies \( g(n) \) is \( O(f(n)) \).

b) \( f(n) + g(n) \) is \( \Theta(\min(f(n), g(n))) \).

c) \( f(n) \) is \( O(g(n)) \) implies \( \lg(f(n)) \) is \( O\lg(g(n)) \), where \( \lg(g(n)) \geq 1 \) and \( f(n) \geq 1 \) for all sufficiently large \( n \).

d) \( f(n) \) is \( O(g(n)) \) implies \( 2^{f(n)} \) is \( O(2^{g(n)}) \).

e) \( f(n) \) is \( O((f(n))^2) \).

f) \( f(n) \) is \( O(g(n)) \) implies \( g(n) \) is \( \Omega(f(n)) \).

g) \( f(n) \) is \( \Theta(f(n/2)) \).

(This is Exercise 3-4 from Chapter 3 of Cormen, Leiserson, Rivest and Stein, *Introduction to Algorithms*, 3rd edition.)