This assignment is due by the start of lecture on September 16, 2009. CLRS refers to the third edition of the textbook.

1. **Collaborators? (5 points)** Clearly indicate your collaborators. If none, state that.

2. **Perspective (15 points)** Suppose that one operation takes one microsecond (= 10^{-6} seconds). With one processor, what is the largest value of \( n \) such that:
   (a) \( \log n \) operations finish in 1 hour
   (b) \( n^3 \) operations finish in 1 day
   (c) \( 2^n \) operations finish in 1 week.

3. **Adding binary integers (20 points)** CLRS 2.1-4. Consider the problem of adding two \( n \)-bit binary integers, stored in two \( n \)-element arrays \( A \) and \( B \). The sum of the two integers should be stored in binary form in an \((n + 1)\)-element array \( C \).
   (a) State the problem formally.
   (b) Write pseudocode for adding the two integers.

4. **Faster insertion sort? (20 points)** CLRS 2.3-6. Observe that the while loop of lines 5-7 of the INSERTION-SORT procedure in Section 2.1 uses a linear search to scan (backward) through the sorted subarray \( A[1 \ldots j - 1] \). Can we use a binary search (see Exercise 2.3-5) instead to improve the overall worst-case running time of insertion sort to \( \Theta(n \log n) \)? Explain why or why not.

5. **Sum to \( x \)? (20 points)** CLRS 2.3-7.* Describe a \( \Theta(n \log n) \)-time algorithm that, given a set \( S \) of \( n \) integers and another integer \( x \), determines whether or not there exist two elements in \( S \) whose sum is exactly \( x \). Briefly justify the running time of your algorithm. If you get stuck, give the best algorithm that you can and analyze its running time.

6. **Order of Growth (20 points)** Answer each of the following with TRUE or FALSE. You do not need to justify your answers. (Note: when dealing with sets like \( O(f(n)) \), \( \Omega(f(n)) \), etc., we use the symbols = and \( \in \) interchangeably.)

   1. \( 200 = O(n) \)
   2. \( 3^n = 2^{O(n)} \)
   3. \( n^n = O(n!) \)
   4. \( \frac{1}{n} = o(1) \)
   5. \( 2n = o(n^2) \)
   6. \( \log(n^2) = \Theta(\log^2(n)) \)
   7. \( n - \log(n) = \Theta(n) \)
   8. \( 2n^5 = \omega(n^5) \)
   9. \( n^2 - 100 = \Omega(n^2) \)
   10. \( 2^n \cdot 2^{2n} = \Omega(2^{3n}) \)