**Algorithms (CSCI 323 & 700)**

**Spring 2016 - Homework #3**

**(Due at the beginning of class on 2/24/2016)\***

1. Using the formal definition of Ω, prove that 3n3 + 2n2 + n = Ω(1000n2 + 2000n + 3000).
2. Using the formal definition of Θ, prove that 100nlogn + 50n = Θ(50nlogn + 100n).

1. For each, state whether it is true or false and offer a brief explanation:
2. if f(n) = O(g(n)) and g(n) = O(f(n)), then f(n) = g(n)
3. if f(n) = Θ(g(n)) and g(n) = Θ(h(n)), then f(n) = Θ(g(n))
4. if f(n) = O(g(n)) and g(n) = O(h(n)), then h(n) = Ω(f(n))
5. if f(n) = ω(g(n)) and f(n) = o(h(n)), then h(n) = Θ(g(n))
6. f1(n) = O(g1(n)) and f2(n) = O(g2(n)), then f1(n) x f2(n) = O(g1(n) x g2(n))
7. Suppose you want to find both the minimum *and* maximum of an array of integers. How might you combine the two separate algorithms for minimum and maximum into one algorithm that is more efficient? What would be the best, average, and worst case of your combined algorithm?
8. As observed in class, if after one pass (iteration) through the array in Bubble Sort there are no swaps, then the array is already sorted. Modify the Bubble Sort algorithm from class to take advantage of this observation. What is the new best, average, and worst case analysis of the modified algorithm?
9. One problem with Bubble Sort is that while large elements move quickly to their final position toward the end of the array, small elements move only slowly to their final position toward the beginning of the array. This phenomenon has been dubbed “Rabbits and Turtles” (since the hare is fast and the tortoise is slow.) Design a variation of Bubble Sort that uses a back-and-forth approach to ensure that both are achieved at equal speed. (Hint: see <https://en.wikipedia.org/wiki/Bubble_sort#Rabbits_and_turtles>.)
10. Explain why T(1) = 0, T(n) = T(n-1) + n-1 aptly describes the number of comparisons for Bubble Sort and then, using the techniques from class, solve the recurrence.
11. Write pseudocode for an algorithm (can be made up) whose time complexity is described by the recurrence T(1) = 1, T(n) = T(n/2) + n.
12. Solve the recurrence in the previous question.

\* Only if you will not be able to attend class on the due date, submit your solutions - *before 6:00 p.m. on the due date* - to the instructor at LT.CS320@yahoo.com