# COP 3503 Honors – Midterm Take Home Exam (Non-Collaborative)

## Due Date: October 23, 2019

1. Solve the following recurrence using the substitution method (10 pts)

   $T(n) = 4T(n/2) + n$, $T(1) = 1$ ( Guess $T(n) = \Theta(n^2)$)

2. Solve the following recurrence relation using the iteration method (find a tight upper bound) (10 pts)

   $T(n) = 4T(n/2) + n^2$

3. Which sorting algorithm would you use in each of the following cases and why: (10 pts)
   a. Sorting *n* floating point numbers that have values between 0 and 1 and 5 significant digits
   b. Sorting *n* floating point numbers in the range of [0,…,100].
   c. Sorting n integers in the range [0,…,*k*] where *k* is much smaller than *n*.
   d. Sorting n floating point numbers that are generated with a uniform distribution in the range [0,…,100]
   e. Sorting *n* integers in the range [0,…, $n^5 - 1$]

4. Use Strassen's algorithm to compute the following product (show all work and intermediate steps) (15 pts)

$$\begin{pmatrix} 1 & 2 \\ 5 & 7 \end{pmatrix} \begin{pmatrix} 4 & 9 \\ 2 & 3 \end{pmatrix}$$

5. (a) You are given a set *S* containing *n* integers in the range [1, …, *k*]. You want to preprocess the input so you can answer queries of the type, "How many elements in *S* fall in the range [*a*,…,*b*]" (where *a,b* are real numbers) as fast as possible. Give both the preprocessing and query answering algorithm and analyze their running time. (15 pts)

   (b) What if instead of integers the set *S* contains floating point numbers? How would we preprocess the set and what would be an efficient algorithm for answering the queries? (10 pts)

6. A thief is in a room that contains n items. Each item has a price *p* and a weight *w* associated with it. The thief can only carry up to *b* pounds (she can either take or leave the item, it is not possible to take a fraction of it). She wants to choose a subset of the items so that the weight of all of them is less than or equal to *b* and her profit is maximized.
    a. Write an algorithm that, given an array W[1…*n*] containing the weights of the items, an array P[1…*n*] containing their prices, and the bound *b*, will output the maximum profit possible. Analyze the time and space requirements of your algorithm (20 pts).
    b. Briefly describe what changes you would make to the above algorithm if you wanted to also compute which items she would take. (10 pts)