CS 590 Midterm Exam

(100 Points)

Collaboration Policy. Exams will be done individually: each student must hand in their own answers. It is acceptable for students to collaborate in understanding the material but not in solving the problems or programming. Use of the Internet is allowed, but should not include searching for existing solutions.

Under absolutely no circumstances code can be exchanged between students. If some code was shown in class, it can be used, but it must be obtained from Canvas, the instructor or the Course Assistants.

Assignments from previous offerings of the course must not be re-used. Violations will be penalized.

Late Policy. No late submissions will be allowed without consent from the instructor. If urgent or unusual circumstances prohibit you from submitting the exam in time, please email me; otherwise the Syllabus late penalty applies.

Deliverable. A single professionally typed pdf or MS Word file on Canvas. **Show your work in detail**, not just answer. No handwritten work or pictures of handwritten work are accepted.

1. (5 Points) Using the very definition of Big-omega notation, prove that n^3 logn is $\Omega(n^3)$. You must use the definition and find the constants in the definition to receive credit.
2. (5 Points) Given that T(n) = 1 if n=0 and T(n) = T(n-1)+ 2^n otherwise; show, by induction, that T(n) = 2^{n+1} -1. Show all three steps of your induction explicitly.
3. (10 Points) Is the bucket-sort algorithm in-place? Why or why not?
4. (16 Points) Describe an efficient algorithm to count the number of nonzero elements of an n by n array A, where in any row of A all nonzero elements come before any zeros in that row.
5. (16 Points) Consider a sequence of n numbers. Describe an efficient algorithm to identify the number that appears the most in the sequence. What is the running time of your algorithm?
6. (16 Points) Given a character string X of length n, describe and **pseudocode** O(n)-time algorithm to construct the set, C, of distinct characters that appear in C, along with the count, f(c), for each c in C of how many times the character c appears in X. You may assume

that the characters in X are encoded using a standard character indexing scheme, like the ASCII system.

7. (16 Points) Suppose a social network, N, contains n people, m edges, and c connected components. What is the exact number of times that each of the methods, **makeSet**, **union**, and **find,** are called in computing the connected components for N using Algorithm 7.2?

8. (16 Points) Develop an algorithm that computes the kth smallest element of a set of n distinct integers in O (n + k log n) time.