CSC 505, Homework 1
Due date: September 4, 9 PM

Homework should be submitted using WolfWare Submit Admin in PDF, or plain text. To avoid reduced marks, please submit **word/latex-formated PDF** file, **NOT scanned writing in pdf format**. Scanned writing is hard to read, takes longer to grade, and produces gigantic files. All assignments are due on 9 PM of the due date. Late submission will result in 10%/40% point reduction on the first/second day after the due date. No credit will be given to submission that are two or more days late. Please try out Submit Admin well before the due date to make sure that it works for you.

All assignments for this course are intended to be individual work. Turning in an assignment which is not your own work is cheating. The Internet is not an allowed resource! Copying of text, code or other content from the Internet (or other sources) is plagiarism. Any tool/resource must be approved in advance by the instructor and identified and acknowledged clearly in any work turned in, anything else is plagiarism.

1. (6 points) Goal: Practice analysis of algorithms. Consider the algorithm represented by the following program fragment.

```
MYSTERY(int x, int n)
{
      int sum=0;
      if x>10 then {
            for (int i=1; i<=n^2; i++) {
                  sum=sum*i;
            }
      } else {
            for (int i=1; i<=n; i++) {
                  sum=sum+i;
            }
      }
      return x + sum;
}
```

a) (2 points) As a function of x and n, what value does MYSTERY(x,n) return?
b) (2 points) Use only ARITHMETIC operations as basic operations, and assume that the size of the input is measured by n. Compute the worst-case running time of MYSTERY.
c) (2 points) Assume that x remains constant while n goes to infinity. Derive a tight, big-Oh expression (dependent on the value of x) for the running time of MYSTERY. Justify your solution.

2. (8 points) *Purpose: Learn about bubblesort, practice running time analysis, learn how loop invariants are used to prove the correctness of an algorithm.* Tip: re-read Section 2.1 in our textbook. Solve problem 2-2 [a-d] on page 40 of the textbook.

3. *Purpose: Practice working with asymptotic notation.* Please solve (12 points) 3-2 on

page 61 , (8 points) 3-4 [a-c] on page 62. For full marks justify your solutions.

4. *Purpose: Practice working with asymptotic notation.* Rank the following functions by order of growth.

a) (5 points) Find an arrangement $f_1, f_2, \ldots, f_n$ of the functions satisfying $f_1 = \Omega(f_2)$, $f_2 = \Omega(f_3)$, ..., $f_{n-1} = \Omega(f_n)$. (Here, lg := logarithm base 2)

n^2, $\log(n!)$, n^{-2}, $e^{2\ln(n)}$, $\ln(n^2)$, 1000, $\mathrm{sqrt}(n)$, $3^{\lg(n)}$

b) (2 point) The order is NOT unique. Which functions can be exchanged, i.e. $f_k \in \Theta(f_{k+1})$.

5. (2 points) *Purpose: Practice working with asymptotic notation.* Assume that a, b, and t are integers, and a,b,t>0. Show $e^{an} \in \omega((bn)^t)$.

6 (8 points) *Purpose: Practice algorithm design and algorithm analysis.* Give pseudocode for a $\Theta(\lg n)$ algorithm which computes a^n, given a and n. Justify the asymptotic running time of your algorithm. Do **not** assume that n is a power of 2. (Here, lg := logarithm base 2)

How to "describe" an algorithm (taken from Erik Demaine)?

Try to be **concise, correct, and complete**. To receive full marks, you should provide (1) a textual description of the algorithm, and, if helpful, pseudocode; (2) at least one worked example or diagram to illustrate how your algorithm works; (3) a proof (or other indication) of the correctness of the algorithm; and (4) an analysis of the time complexity (and, if relevant, the space complexity) of the algorithm. **Remember that, above all else, your goal is to communicate.** If a grader cannot understand your solution, they cannot give you appropriate credit for it.