COT 6405 Analysis of Algorithms

Homework 1

DUE: Sunday September 10 at 11:55 PM

- Please remember that all submissions must be typeset. Handwritten submissions will NOT be accepted. These must be uploaded to SCIS moodle in PDF format only.
- Please remember to type your name on top of your submission.

1. (20 points)
   Rank the following functions by increasing order of growth. That is, find any arrangement $g_1, g_2, g_3, g_4, g_5, g_6$ and $g_7$ of the functions satisfying $g_1 = O(g_2), g_2 = O(g_3), g_3 = O(g_4), g_4 = O(g_5), g_5 = O(g_6)$ and $g_6 = O(g_7)$.

   
   \[
   \begin{align*}
   f_1(n) &= (\log n)!, & f_2(n) &= 2^{2^{n+1}}, & f_3(n) &= (\log n)^{\log n}, & f_4(n) &= \binom{n}{3}, \\
   f_5(n) &= 2^{2^n}, & f_6(n) &= \log \log n, & f_7(n) &= (\log n)^{1/2}, & f_8(n) &= n!
   \end{align*}
   \]

2. (20 points) Solve the following recurrence equations. As usual, you may assume that $T(n) = O(1)$ when $n$ is smaller than a fixed constant. Below $c$ is an absolute constant.

   (a) (5 points) $T(n) = 4T(n/2) + cn$.
   (b) (5 points) $T(n) = T(n-4) + cn$.
   (c) (5 points) $T(n) = 3T(n-4) + c$.
   (d) (5 points) $T(n) = 4T(n/2) + n^3$ (Use Master Theorem)

3. (20 points) Design an efficient algorithm that computes the product of two $n$-bit integers in $o(n^2)$ time.

4. (20 points) For any two sets $X$ and $Y$ of integers, the Minkowski sum $X + Y$ is the set of all pairwise sums $\{x + y : x \in X, y \in Y\}$.

   (a) Design an algorithm that computes the number of elements in $X + Y$ in $O(n^2 \log n)$ time.
   (b) Design and analyze an algorithm that computes the number of elements in $X + Y$ in $O(M \log M)$ time, where $M$ is the largest absolute value of any element of $X \cup Y$.

5. (20 points)

   (a) Design an algorithm that determines whether a given set of $n$ integers contains two elements whose sum is zero, in $O(n \log n)$ time.
   (b) Design an algorithm that determines whether a given set of $n$ integers contains three elements whose sum is zero, in $O(n^2)$ time.
   (c) Let $S \subset \{-100n, -100n + 1, \ldots, 200n - 1, 200n\}$ be a set of $n$ integers. Design an algorithm that determines whether $S$ contains three elements whose sum is zero, in $O(n \log n)$ time.