CS 4413 Algorithm Analysis **Homework 3**

Fall 2019 **Due:** Sep 12, 2019

Please answer each of the following problems.

Note: For all problems, if you include pseudocode in your solution, please also include a brief english description of what the pseudocode does.

1. (a) Prove T(n) is a polynomial of n with degree 2 for selection sort. Plot T(n) vs n for $2 \leq n \leq 50$. Compute $\lim_{n\to\infty} \frac{T(n)}{n^2}$. (5 pt)

   (b) let $f_1(n) = \frac{3}{4}n^2$, $f_2(n) = \frac{1}{4}n^2$. Plot $f_1(n), T(n), f_2(n)$ on the same plot for $2 \leq n \leq 50$. For what values of n do you observe the following relation:
   $f_2(n) < T(n) < f_1(n)$ (5 pt)

2. Assume we have five asymptotically non-negative functions $f_1(x), f_2(x), f_3(x), f_4(x), f_5(x)$. Prove $\max(f_1(x), f_2(x), f_3(x), f_4(x), f_5(x)) = \theta(f_1(x) + f_2(x) + f_3(x) + f_4(x) + f_5(x))$. (10 pt)

3. We have an integer 'z' and an array A with 'k' integers. Write a pseudocode with $\theta(nlogn)$ to determine if there are two integers in A whose summation is equal to integer 'z'. You need to explain your pseudocode clearly and analyze why it takes $\theta(nlogn)$. (10 pt)

4. Solve the following recurrences using method of substitution:

   (a) $T(n) = T(\lfloor \frac{n}{2} \rfloor) + 1$, where T(1) = 1 and $n = 2^k - 1$ (5 pt)

(b) $T(n) = T(\sqrt{n}) + 1$, $n = 2^{2^k}$, T(2) = 1 (2.5 pt)

(c) $T(n) = 3T(n/2) + 8n$, $n = 2^k$, T(1) = 1 (2.5 pt)

5. (a) Is $5^{5n} = O(5^n)$? Is $5^{n+1} = O(5^n)$? (5 pt)

   (b) Which is asymptotically larger: $log(log^*n)$ or $log^*(logn)$? (5 pt)

6. In the HW2, we saw how insertion sort runs better than merge sort with the help of constant factors (when array size is smaller). Thus, it makes sense to use insertion sort within merge sort when subproblems become sufficiently small. Consider a modification to merge sort in which n/z sublists of length z are sorted using insertion sort and then merged using the standard merging mechanism, where z is a value to be determined.

   (a) Show that the n/z sublists, each of length z, can be sorted by insertion sort in $\theta(nz)$ worst-case time. (5 pt)

   (b) Show that the sublists can be merged in $\theta(nlog(n/z))$ worst-case time. (5 pt)

   (c) Given that the modified algorithm runs in $\theta(nz + nlog(n/z))$ worst-case time, what is the largest asymptotic (θ-notation) value of z as a function of n for which the modified algorithm has the same asymptotic running time as standard merge sort? (5 pt)

   (d) Find the value z in practice ? (5 pt)

   Bonus Question : Write a program (R/Python/MATLAB) and implement the above idea. You need to submit your program separately in canvas along with the PDF. (5 pt)