COSC 030, Fall 2016: Problem Set $\#3$

Assigned: Tuesday, 9/20.
Due: Thursday, 9/29, at the beginning of class (hand in hard copy). (***Notice Thursday Due Date.***)
Lectures Covered: Week 4.
Academic Integrity: You must work alone on the problem set and not consult outside sources (with the exception of the professor and teaching assistants). See the syllabus for details on the academic integrity policy for problem sets.

Problems (2 pages)

1. Consider the *Lonely Max* problem which provides an algorithm a sequence $a_1, a_2, ..., a_n$ of natural numbers. Let a_{max} equal the maximum value in the sequence (i.e., $a_{max} = \max_{1 \leq i \leq n}\{a_i\}$). The problem requires the algorithm to output *true* if there exists exactly one position i in the sequence such that $a_i = a_{max}$, otherwise it must output *false.*

   What is wrong with the solution proposed below?

   LonelyMaxCheck$(a_1, a_2, ..., a_n)$

   $max \leftarrow$ the largest value in the input sequence

   $count \leftarrow$ the number of positions in sequence that have value max

   if $count = 1$ **then return** $true$

   else return $false$

2. Consider the *Sum Check* problem which provides an algorithm a sequence $a_1, a_2, ..., a_n$ of natural numbers, where n is an even number of size at least 2, as input. It requires the algorithm to return $true$ if $\sum_{i=1}^{n/2} a_i = \sum_{j=(n/2)+1}^{n} a_j$ and otherwise return $false$.

   Describe a correct algorithm for the *Sum Check* problem that *writes* to at most two variables and does *not* use a **for** loop. That is, your algorithm can have at most two variables whose values might change at some point during its execution.

   (*Note*: you can reference sizes like $n/2$, $n/2 + 1$ and/or n in your pseudocode without this counting as a write to a variable.)

3. What is the exact (i.e., not asymptotic) worst-case step complexity of your algorithm? Provide an explanation for how you arrived at this value.

4. For each of the following five statements, specify if it is *true* or *false*. (No explanation required.)

   (a) $19n^2 + 1000n^3$ is $\Theta(n^2)$
   (b) $2n + 20$ is $O(n^2)$
   (c) $2^{(\log_2 n)+1}$ is $\Omega(n \cdot \log n)$
   (d) 2^n is $O(n^2)$
   (e) $n/100$ is $\Omega(\sqrt{n})$

5. Consider the function $f(x) = 7x^2 + x + 19$.

   (a) Define a function $g(x)$ such that $g(x)$ is $O(f(x))$ but $g(x)$ is *not* $\Omega(f(x))$.

   (b) Define a function $h(x)$ such that $h(x)$ is $\Omega(f(x))$ but $h(x)$ is *not* $O(f(x))$.

   (c) Define a function $j(x)$ that is *different* than $f(x)$ but is also $\Theta(f(x))$.