COSC 030, Fall 2019: Problem Set $\#3$

Assigned: Thursday, 9/19.
Due: Thursday, 9/26, at the beginning of class (hand in hard copy).
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 (two 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., $a_{max} = \max_{1 \le i \le n}\{a_i\}$). 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, for even number $n \ge 2$, 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 makes use of a single **for** loop, and only writes to a single variable (not counting the index of the **for** loop). You can do arithmetic in the sequence subscripts without counting the operation as a write; e.g., referencing a_{i+1} , where i is the **for** loop index, does not count as a write.

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. (You can decide what constants you want to use for individual operations.)

4. Prove your algorithm from problem 2 is correct. If you were unable to come up with a solution to problem 2 that used only a single variable, you may write out a new solution here that uses multiple variables and prove that this new version is correct.

   (Hint: for this problem, proving correctness means you must show two things: if it outputs $true$, the relevant sums are equal, and if it outputs $false$, the relevant sums are not equal.)

More problems on next page...

5. 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 $\Omega(\sqrt{n})$

6. 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 $\Theta(f(x))$.