COSC 030, Fall 2015: Problem Set $\#3$

Assigned: Tuesday, 9/22.
Due: Tuesday, 9/29, 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

1. 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 returns $false$.

   Why is the following solution to the *Sum Check* problem not a valid computer algorithm?

   DoubleSumAlg$(a_1, a_2, ..., a_n)$
   $x \leftarrow$ **SumValues**$(a_1, ..., a_{n/2})$
   $y \leftarrow$ **SumValues**$(a_{n/2+1}, ..., a_n)$
   if $x = y$ **then return** $true$
   else return $false$

2. Describe a correct algorithm for the *Sum Check* problem that uses *at most two* variables.

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 $f(x) = 25x + 1000$.

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