COSC 030, Fall 2014: Problem Set #3

Assigned: Tuesday, 9/16.
Due: Tuesday, 9/23, 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 *Two Search* problem which provides an algorithm a sequence $a_1, a_2, ..., a_n$ of natural numbers as input, and requires the algorithm to return $true$ if at least one a_i value equals 2, and otherwise requires the algorithm to return $false$.

   Why is the following solution to the *Two Search* problem not a valid computer algorithm?

   SplitSearch$(a_1, a_2, ..., a_n)$
   $x \leftarrow$ **Contains2**$(a_1, ..., a_{\lfloor n/2 \rfloor})$
   $y \leftarrow$ **Contains2**$(a_{\lfloor n/2 \rfloor + 1}, ..., a_n)$
   if $x = true$ **or** $y = true$ **then return** $true$
   else return $false$

2. Describe a correct algorithm for the *Two Search* problem.

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 statements, specify if it is *true* or *false*.

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

5. Consider the function $f(x) = 25x$.

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