COSC 240, Fall 2017: Problem Set #2

Assigned: Wed, 9/20/17.
Due: Wed, 9/27, at the beginning of class (hand in hard copy).
Lectures Covered: 5 to 6.
Academic Integrity: You can work with other people in the class but you must write up your own answers in your own words. You can also use the textbook and talk to the professor. You may not use any other resources (e.g., material found online) or talk to people outside the class about these problems. See the syllabus for details on the academic integrity policy for problem sets.

Problems

1. Describe with pseudocode a recursive divide-and-conquer algorithm that finds the largest value in an array A of numbers. In more detail, your algorithm should take as arguments an array A, and two integers p, r, such that $1 \leq p \leq r \leq length(A) = n$, and return the largest value in $A[p...r]$.

2. Describe the step complexity of your solution to problem 1 with a recurrence. Solve the recurrence using the master method.

3. Describe clearly and concisely the problem solved by the below divide-and-conquer algorithm (you can assume the input A is an array containing numbers, and that $1 \leq p \leq r \leq length(A)$):

   ALG(A, p, r)
   if $p = r$ **then**
   return $A[p]$
   else
   $q \leftarrow$ a value chosen with uniform randomness from $\{p, p+1, ..r-1\}$
   $a \leftarrow$ ALG(A, p, q)
   $b \leftarrow$ ALG$(A, q+1, r)$
   return $a + b$

4. Let X be a random variable that describes the number of times the addition operator $(+)$ is applied in a given execution of ALG. Prove a Θ bound on the value of X (an exact bound would be fine here as well). Notice, this a bound on X, not $E[X]$. In other words, you are providing a bound on the size of X that holds regardless of the random choices. In proving your bound you may use the graph theory fact that a full binary tree with ℓ leaf nodes has $\ell - 1$ internal (non-leaf) nodes.

5. Use your answer to the previous problem to bound the step complexity of ALG without having to calculate an expectation.