- See Practice Problems Sets 1 and 2 for instructions.
- Explicitly state all the assumptions that you use in your pseudocode and analysis.

1. Refer to the sorting algorithm in your solution to Problem 8 in Problem Set 2.
   (a) Obtain an *asymptotically tight* (that is, of the form $T(n) = \Theta(f(n))$, for some function f) bound on the *worst-case* running time of this algorithm on an input list of size n.
   (b) Obtain an asymptotically tight bound on the *best-case* running time of this algorithm on an input list of size n.

2. Read up on what a palindrome is, if required. Your solution to parts (c) and (d) must consist of deriving recurrence relations and solving them using the "estimate and verify" method that we saw in class.
   (a) Write the pseudocode for a **recursive** function IsPalindrome($string$) that checks if the given string is a palindrome. (*Hint:* You already did something like this for Problem Set 1.)
   (b) Prove using induction that your pseudocode of part (a) is correct. (*Hint:* Ditto.)
   (c) Obtain an asymptotically tight bound on the worst-case running time of your algorithm of part (a).
   (d) Obtain an asymptotically tight bound on the best-case running time of your algorithm of part (a).

3. Refer to your algorithm FindLargest from Problem Set 1. For each part below, your analysis must consist of deriving recurrence relations and solving them using the "estimate and verify" method that we saw in class.
   (a) Obtain an asymptotically tight bound on the worst-case running time of your algorithm, in terms of the number of numbers in the input.
   (b) Obtain an asymptotically tight bound on the best-case running time of your algorithm, in terms of the number of numbers in the input.

4. Read up the definition of the Longest Common Substring problem.
   (a) Write the pseudocode for an algorithm that solves this problem for two input strings x, y, starting with a *recursive* solution for the following simpler version: Given (x, y, i, j, r) as input where i, j, r are integers, check whether the r-length substrings of x, y that start at indices i, j, respectively, are identical. That is, whether $x[i]x[i+1] \cdots x[i+r-1]$ is identical to $y[i]y[i+1] \cdots y[i+r-1]$.

(b) Let m, n be the lengths of inputs x, y, respectively. Analyze your algorithm of part (a) to obtain asymptotic upper and lower bounds on the time it takes to solve Longest Common Substring, in terms of m and n.

5. Prove the following statement using the definition of the $\mathcal{O}()$ notation that we saw in class:

Claim

Let $f(n)$, $g(n)$, and $h(n)$ be functions from non-negative integers to real numbers. If $f(n) = \mathcal{O}(g(n))$ and $g(n) = \mathcal{O}(h(n))$, then $f(n) = \mathcal{O}(h(n))$.

6. Prove or disprove:

(a) $3n = \mathcal{O}(2n)$

(b) $\log_2 3n = \mathcal{O}(\log_2 2n)$

(c) $2^{3n} = \mathcal{O}(2^{2n})$

(d) $\sqrt{n} = \mathcal{O}(n^{\sin n})$

(e) $n^{\sin n} = \mathcal{O}(\sqrt{n})$

(f) $n^{\log_2 \log_2 n} = \mathcal{O}(2^{\sqrt{\log_2 n}})$

(g) $2^{\sqrt{\log_2 n}} = \mathcal{O}(n^{\log_2 \log_2 n})$