COMP285: Analysis of Algorithms
North Carolina A&T State University
Dr. Allison Sullivan

Homework #1
Due: August 31, 2018 (*in class quiz*)

Homework #1

You should try to solve these problems by yourself. I recommend that you start early and get help in office hours if needed. If you find it helpful to discuss problems with other students, go for it. **You do not need to turn in these problems. The goal is to be ready for the in class quiz that will cover the same or similar problems.**

Problem 1: Asymptotic Time Complexity

Consider each of the following pairs of functions. For each pair, either $f(n) = O(g(n))$, $f(n) = \Omega(g(n))$ or $f(n) = \Theta(g(n))$. Determine which of these three options best captures the relationship and (briefly) explain or demonstrate why.

(a) $f(n) = \log n^2$; $g(n) = \log n + 5$

(b) $f(n) = \sqrt{n}$; $g(n) = \log n^2$

(c) $f(n) = \log^2 n$; $g(n) = \log n$

(d) $f(n) = n$; $g(n) = \log^2 n$

(e) $f(n) = n \log n + n$; $g(n) = \log n$

(f) $f(n) = 10$; $g(n) = \log 10$

(g) $f(n) = 2^n$; $g(n) = 10n^2$

(h) $f(n) = 2^n$; $g(n) = 3^n$

Problem 2: Algorithm Analysis

Answer the following questions based on the following psuedocode for the function "foo":

Algorithm 1: foo

```
mystery()
foreach i ←1 to n do
    mystery()
    foreach j ←1 in n do
        if i ≤ j then
            mystery()
```

A. Determine the exact number of times mystery() is called in terms of n.

B. Assume the mystery function is in $O(log(n))$. Determine the complexity of "foo" in O-notation.

Problem 3: Master Method

Use the master method to give a tight asymptotic bound for each of the following recurrences.

1. $T(n) = 8T(n/2) + \Theta(n^3 \log n)$
2. $T(n) = 3T(n/2) + \Theta(n)$
3. $T(n) = 3T(n/2) + \Theta(n^2)$
4. $T(n) = 16T(n/2) + \Theta(n^3 \log n)$
5. $T(n) = T(9n/10) + \Theta(n)$

Problem 4: Divide and Conquer

Suppose you are given a sorted sequence of *distinct* integers $\{a_1, a_2, \ldots a_n\}$. Give an $O(\log n)$ algorithm to determine whether there exists an index i such that $a_i = i$. For example, in $\{-10, -3, 3, 5, 7\}$, $a_3 = 3$; there is no such i in $\{2, 3, 4, 5, 6, 7\}$. Write the recurrence for your algorithm and show that its recurrence solves to $O(\log n)$ (e.g., using the Master Method, the iteration method, or a recursion tree).