CS325: Analysis of Algorithms, Fall 2017

Practice Assignment 1*

Due: Tue, 10/3/17

Homework Policy:

1. Students should work on practice assignments individually. Each student submits to TEACH one set of *typeset* solutions, and hands in a printed hard copy in class or slides it under my door before the midnight of the due day. The hard copy will be graded.
2. Practice assignments will be graded on effort alone and will not be returned. Solutions will be posted.
3. The goal of the assignments is for you to learn solving algorithmic problems. So, I recommend spending sufficient time thinking about problems individually before discussing them with your friends.
4. You are allowed to discuss the problems with others, and you are allowed to use other resources, but you *must* cite them. Also, you *must* write everything in your own words, copying verbatim is plagiarism.
5. More items might be added to this list. ☺

Problem 1. For each of the following, indicate whether $f = O(g)$, $f = \Omega(g)$ or $f = \Theta(g)$.

(a) $f(n) = 12n - 5$, $g(n) = 1235813n + 2017$.

(b) $f(n) = n \log n$, $g(n) = 0.00000001n$.

(c) $f(n) = n^{2/3}$, $g(n) = 7n^{3/4} + n^{1/10}$.

(d) $f(n) = n^{1.0001}$, $g(n) = n \log n$.

(e) $f(n) = n6^n$, $g(n) = (3^n)^2$.

Problem 2. Prove that $\log(n!) = \Theta(n \log n)$. (Logarithms are based 2)

Problem 3. Write a recursive algorithm to print the binary representation of a non-negative integer. Try to make your algorithm as simple as possible. Your input is a non-negative integer n. Your output would be the binary representation of n. For example, on input 5, your program would print '101'.

*Some problems are from Jeff Erickson's lecture notes. Looking into similar problems from his lecture notes on recursion is recommended. Other problems are from the book "Introduction to algorithms" by Coremen, Leiserson, Rivest, and Stein. Fore more on asymptotic running time analysis, see Chapter 2 of this book.

Problem 4.

(a) Read tree traversal from wikipedia: `https://en.wikipedia.org/wiki/Tree_traversal`, the first section, Types.

(b) Recall that a binary tree is *full* if every non-leaf node has exactly two children. Describe and analyze a recursive algorithm to reconstruct an arbitrary full binary tree, given its preorder and postorder node sequences as input. (Assume all keys are distinct in the binary tree)

The following problems are for practice. Do *not* submit solutions for them.

Practice Problem A. Write a recursive algorithm to count the number of binary strings of length n with no consecutive 1's. Your input is a non-negative integer n. Your output should be the number of binary strings of n bits with no consecutive ones. For example, on input 1, your algorithm returns 2 ('1', '0'), on input 2, your algorithm returns 3 ('00', '01', '10'), and on input 3 your algorithm returns ('000', '010', '100', '001', '101').

Practice Problem B. Collatz sequence starting at an integer n is defined as follows. Start with an integer n. In each step, if n is even divide it by two (i.e. $n = n/2$), if it is odd multiply it by three and add 1 to it (i.e. $n = 3n + 1$). Write a "recursive" algorithm to generate Collatz sequence. Can you show that your algorithm ends? What element of recursion is missing here? See `https://en.wikipedia.org/wiki/Collatz_conjecture`.

Practice Problem C. Let $f(n)$ and $g(n)$ be nonnegative functions. Use the definition of Θ-notation to prove that $\max(f(n), g(n)) = \Theta(f(n) + g(n))$.

Practice Problem D. Continuation of Problem 4.

(c) Recall that a binary tree is *full* if every non-leaf node has exactly two children. Describe and analyze a recursive algorithm to reconstruct an arbitrary full binary tree, given its preorder and postorder node sequences as input.

(d) Describe and analyze a recursive algorithm to reconstruct an arbitrary binary tree, given its preorder and inorder node sequences as input.

(e) Describe and analyze a recursive algorithm to reconstruct an arbitrary *binary search tree*, given only its preorder node sequence. Assume all input keys are distinct.

Practice Problem E. Use induction to prove the following facts.

(a) $1 + 2 + \ldots + n = \frac{n(n+1)}{2}$.

(b) $1 + c + c^2 + \ldots + c^n = \frac{c^{n+1}-1}{c-1}$, for any $c > 0$.