**CSC 379 Homework 1**

**UM-Flint, Winter 2019**

Copyright © 2019, Dr. Matthew Spradling. All rights reserved.

**Due Date:** Thursday, 1/24/2019, 11:59 PM EST. Late assignments not accepted without an approved excuse.

**Submission Format:** Submit typed as a pdf file. Use the Attach File option through Blackboard, on this assignment.

**I will not accept hand-written assignments.**  PDF submissions **may not include images of typed text**. Any text which is contained within an image file **will not be graded**. Yes, I can tell the difference. You may use image files for any diagrams you need to draw, however.

**Scoring:**

Maximum Points Possible: 100

**Group Member Rules:** You may have **unlimited** partners at no cost**.**

Each person must submit his or her own copy of the assignment to receive credit. You are not required to have a partner. You can freely chat on the discord server about questions and solutions. If someone (other than the instructor) is particularly helpful to you, please cite them on your assignment.

However, and this is important: **Please do not simply give solutions away.** The reason is that if someone does not understand the solutions, does not practice on their own, etc. this means they will A) not develop a mastery of the material and B) **probably not do well on the exams.**   
  
This is an experiment. I want to see how it goes letting you work together more freely. You are adults and I want to try to treat you as such. So, to that end, on this assignment I will trust that you will endeavor to understand things rather than just get quick solutions without understanding them etc. If the exams show a lower level of understanding on this part of the material I will probably go back to fewer partners. Impress me with how you handle the freedom!

**General Homework Rules:**

**You may NOT use outside sources to obtain solutions to homework problems. You may NOT look for answers on the Internet.**You may read outside sources to help you understand the general material we are covering in the course, but you may not use outside sources to solve the homework problems. "Outside sources" includes other students who are not members of your group. You may talk to other members of the class about the course content, but you may not use other students' solutions to the homework unless they are a member of your group. Similarly, you may not provide solutions to other students unless they are a member of your group. You may not have a member of your group who is not a member of your sections of the course this semester. If you have any doubts about what is allowed, ask your professor.

I will not adjust your grade on an exam or assignment based upon good performance on other homeworks, good performance in other classes, or other factors such as “needing an A.”

**Advice:**

Remember that you are responsible for everything you turn in with your name on it. Use your best judgment on who to work with if you have a partner. If your partner plagiarizes, you have also plagiarized. If there are any issues, please inform me.

Do not try to avoid doing the homework. The minimum penalty for plagiarism is a 0 on the assignment. If plagiarism occurs, and I do not catch it, you can be certain that it will show up in your score on the exams.  Please do not risk it.  You should know at this point that not doing the assignments will result in your being unable to score well on your exams.

Please listen to me on this.  I want you to do well in the course.   Homework is designed to help you achieve the level of mastery needed. Please put the time in on the assignments and you will do better. Please come to office hours, email me, start early, and get help from me if you have trouble.

Helpful shortcuts for the equation mode in Word:

\Theta (Big Theta) produces:   
\O (Big Oh) produces:   
\Omega (Big Omega) produces:   
\o (little oh) produces:   
\omega (little omega) produces:   
n^a (exponent) produces:

n\_a (subindex) produces:   
log\_a b (log) produces:   
a/b (fraction) produces:   
\sum (summation) produces:

\cup (union) produces:

\cap (intersection) produces:

Combine with ()’s to get more complex terms. Examples:

n^(log\_(b-1)a^b) produces: J

n^log\_b-1 a^b produces: LL

(n-1)/(n^2+2) produces: J

n-1/n^2+2 produces: LL

Some logarithmic properties for b>1, a>0, c>0, where a and c are both real numbers. (There are more in the textbook, as you

Please see Appendix C for more information on Probabilities, Expected Value, Expected Value of Random Variables, Counting, Combinations, etc.

Please see Appendix A for more information on Summations / Geometric series, etc.

**Show your work for calculations you perform.**

**Special for calculating summation solutions:** Once you have determined the appropriate summation, you are free to either calculate its solution or to reference a solution. You may reference a book, an online source such as Wikipedia, or to use Wolfram Alpha or some other calculator for this part only. You must cite your source. This is only for **solving** summations, **not** for determining what the summations should be! I would like you to be able to solve summations, but that is not what I am trying to evaluate here.

**Example**: If you found a complexity to be:

Then you could say:

Citing Introduction to Algorithms (Cormen et. al), page 1147, A.5, we have that:

Therefore:

**Homework problems are on the following pages.**

Problem 1) (40 points) (Solving Recurrences)

Solve the following recurrences **for positive *n* values** using the stated method.

1A) using Master Method

Therefore,

1B) using Master Method

Therefore,

1C) using Master Method

Therefore,

1D) using Recursion Tree Method

n

n/4

n/16

n/2

n/8

n/8

n/4

n/64

n/32

n/32

n/16

n/8

n/16

n/16

n/32

The rate of change in the denominator of the summation is higher than the rate of change in the numerator, which will make the summation a constant term. Therefore

Problem 2) (30 points)

Given a recurrence of the form:

Show that the height of the recursion tree is equal to .

The base cases for the recursive function is when T(1) = 1. Therefore, when (n/b) = 1 the recursive function has reached it base case and does no further recursion calls.

Now looking at the function the variable a, it only determines the amount of children the per recursion call, it does not denote the height of the tree.

So, the height of the tree is determined by T(n/b). Knowing that the recursion call has reached its base case when T(n/b) = 1 we now must figure out how many times do you have to drop down a layer in the tree and multiply b by itself in order to for it to equal n, which will mean that it equal 1 which is the base case for the recursive function. This can be shown as were x is the number of layers you have to add and multiple b by itself. And to solve you can change the form and use logarithms to get .

Example

Top layer , Second Layer … and so forth until

Problem 3 (30 points)

Suppose you have a polynomial. This can be represented as a summation

z

where is a vector of coefficients.

We want an efficient function to compute for any given constant and a coefficient vector of size .

**Naïve Approach:** Represent as a coefficient function:

**Horner’s Rule Approach:** Re-write the function as:

**Problem:** Write pseudocode for a recursive algorithm to compute for coefficients in the array and a given constant using the Horner’s Rule Approach. Evaluate the time complexity as an asymptotic bound.

//Preconditons: n coefficients, a is the array of coefficients, x is the constant, and current is the beginning index (0 to n-1)

//Example: horner(5,a,2,0)

Float horner(n, a, x, current) {

If (n-1 == current) {

Return a[current];

}

Return a[current] + x \* horner(n,a,x,current+1);

}

1

….

n-2

n-1

n

1