CMSC 201 Final Exam Study Guide

This guide covers only the materials reviewed that are not on the previous study guides.

The final exam ***is cumulative***. That is, you are responsible for demonstrating any of the skills on the two previous study guides as well as those listed here.

You are going to be expected to evaluate, write, and examine Python code to demonstrate the skills covered in this study guide. Additionally, there will be "normal" questions (multiple choice, short answer, etc.) where you will demonstrate your understanding of the material.

Go over your previous exams and make sure you understand what responses you should have given to get full credit.

The order, length or depth of a section in this study guide is ***no indication*** *of the relative importance of that topic or its likelihood to appear on the exam.* i.e. Don't ignore the short sections!

This study guide is *not finalized* until the last session of the course.

Importing and Modularity

- Explain what a library is, what it is used for, and who is intended to use it.
- Explain the purpose of the import keyword
    - You will NOT be given examples of import in use and asked to explain the details of how they execute

Recursion

- Define recursion, recursive call, and base case
- Given a problem, redefine it recursively
- Implement a recursive function
- Define the call stack
- Explain what problems are best solved with recursion or iteration

Sorting and Searching

- *Explain* the problem of sorting a list
- *Explain* the problem of searching a list
- *Define* the following sorting algorithms and apply them to an input list:
    - Bubble sort

    - Selection sort
    - Quick sort
- *Define* the following searching algorithms and apply them to a input list and target value:
    - Linear/Sequential search
    - Binary search
- *Explain* why binary search only works for lists that are already sorted
- **NOTE:** No dancing will be required before, during or after the exam

Asymptotic Analysis

- *Define* big o and big omega in terms of best and worst case performance
- *Define* big theta in terms of big o and big omega
- *Rank* the following functions from slowest growing to fastest growing (i.e. fastest to slowest)
    - 1 (constant)
    - $\log_2 n$ (logarithmic)
    - n (linear)
    - $n * \log_2 n$ (“n log n”)
    - n^2 (quadratic)
- *Analyze and argue* the best and worst performance of the following algorithms as functions of n:
    - Bubble sort
    - Selection sort
    - Quick sort
    - Linear/Sequential search
    - Binary search
- *Define* log in terms of division

Data representation

- *Define* the decimal, binary and hexadecimal counting systems
- *Convert* a number from any one counting system to another counting system
- NOTE: you will NOT be permitted to use a calculator, but arithmetic will be reasonable to solve by hand

And remember, YOU GOT THIS!