CSSE230 Exam 1 SRT Review Session

The exam will cover the following topics:

- On the written part (60-70% of exam):
    - Finding exact and big-theta runtime of code snippets, knowing when and how to use a loop index table and summations.
    - More with big O, Theta, and Omega: true/false, using definitions to do proofs
    - ADT/Collections: Choosing an ADT to solve a given problem and knowing its runtime
    - Growable Arrays
    - MCSS
    - Something from the assignments, like binary search.
- On the programming part (30-40%):
    - Implementing an ADT using an array, nodes, or another ADT, or
    - Writing an efficient algorithm to solve a simple array-based problem

Section 1. True or False:

Answer the following questions by circling either (T)rue or (F)alse. A statement is only considered true if it is **always** true, and a statement can be considered false if there is a **single counterexample**.

a. ***T*** F if F(N) is O(N) than F(N) is O(N^2)

b. ***T*** F ln(N)/ln(5) is Θ($\log_2$(N))

c. T ***F*** Removing an item from the beginning of an ArrayList<T> is Θ(1)

d. T ***F*** *All* linked lists have an runtime of Θ(1) when adding at the end of the list.

e. T ***F*** An empty tree has a height of 0

Section 2. ADT Identification and Implementations:

For the following examples, supply an ADT, and Implementation, and an operational runtime with explanation.

a. An author, Brett Gravelson, is writing a huge epic book with hundreds of characters. Their publisher has asked them to figure out just how many sequels there will be. The author in an attempt to figure this out has compiled and ordered all of the major story events he has planned into a document. This document is getting too large and he wants to switch to a digital solution. He needs an application that when given a range of numbers it retrieves a range of events. For example when run with (3, 5) that application would get the third, fourth and fifth events.

   ADT: ***List***
   Implementation: ***ArrayList***
   Given one number, runtime for returning a story event: ***Theta(1)***
   Explanation: ***ArrayList.get() is an O(1) operation***

b. Brett Starson has told his publisher that he thinks he will be writing 41 novels in his series. The publisher is excited and stressed about this prospect and they want to help Brett schedule his various tasks if he is to complete these books before he retires. Every book will require 3-5 drafts, a multi stage editorial process, and a book tour for the publisher to consider the book complete. They want an application that schedules these tasks whenever the publisher is confident that they are on schedule. The first task that is scheduled should be the first one that the author sees, and the last one that gets scheduled should be the last thing that the author sees.

   ADT:***Queue***
   Implementation:***Linked List with a Tail Pointer***
   Given a task, runtime for the publisher to add it too the schedule: ***Theta(1)***
   Explanation: ***This really depends on the implementation, in this case it makes sense to be using a tail pointer.***

Section 3. Exact Runtimes and Big Theta Estimates

For each of the following give an exact number and theta estimate in terms of n for the number of times that **sum++** is run.

```
for (int i = 0;i < n * n;i += n) {
      for (int j = n;j > 0;j --) {
            sum++;
      }
}
```

Exact: ***N^3*** Big-theta: Θ(***N^3***)

```
// Assume n is a power of 3
for (int i = n; i > 0;i /= 3) {
      for (int j = 0;j < i; j ++) {
            sum++;
      }
      sum++;
}
```

Exact: ***Sum of (i = 0 to log$_3$n of 3^i) + log$_3$n*** Big-theta: Θ(***N * log(N)***)

Section 4. Maximum Contiguous Subsequence Sum

a. What is the MCSS for {-1, 3, 4, -7, 9, 8, 5, -5, 7, -12, -11}

   Answer for part (a) ***24***

b. If we use the discussed O(N) algorithm, how many times will the sequence reset to zero (not including the initial value)?

   Answer for part (b) ***2 or 1***

Section 5. Big Theta Proof

Show that the following equations match the given big theta values. This will involve selecting two pairs of constants to form Big O and Big Ω proofs.

a. Show that $f(n) = 4n^2 + \log_2(n) + 4 \in \Theta(n^2)$

   $4n^2 + \log_2(n) + 4 <= 4n^2 + n^2 + 4n^2 = 9n^2$
   I'll pick for c_O 9, and for n_O 2

   $4n^2 + \log_2(n) + 4 >= n^2$
   I'll pick for c_Ω 1 and for n_Ω 1

b. Show that $f(n) = \log_2(2^n) + n^3 + n^2/100 \in \Theta(n^3)$

   Very similar, it's still a polynomial time algorithm.