*COP3503 Computer Science 2*
*Dr. Andrew Steinberg*
*Fall 2021*

# Programming Assignment 1
# The Winning Ticket and Deadline Constraints
# Due: 9/27/2021 at 11:59pm

*Objective: Students will apply concepts of Brute Force and Divide and Conquer in this programming assignment.*

**Problem Scenario:** You saw on the news that the Super-Duper Lotto Company is doing a once in a lifetime lotto draw of 1 billion dollars. You decided to take the biggest risk and invest in buying 100000 unique lotto tickets. Later that evening, you watch the news to witness the lotto draw. The numbers are called one-by-one and you find out that you are the lucky winner. Now that you are the lucky winner, the company has a set deadline for the winner to present the ticket in order to claim the prize. If the recipient doesn't present the winning ticket by the deadline, the recipient forfeits the money prize. The recipient will now have to go through the entire set of tickets purchased to find the winner. The recipient knows that the tickets are sorted when they were purchased.

**Assignment Description:** In this programming assignment, you will need to derive two solutions. One solution must run in $O(n)$ time and the other must run in $O(\log n)$ time where $n$ is the number of tickets. This programming assignment must be completed in the Java programming language. Any other language used will result in score of 0 on the assignment.

For this assignment, you must follow these **requirements**.

1. You cannot use any predefined data structure class (such as ArrayList). Using other predefined data structure classes will result in point deductions.
2. You must use the Array as the data structure to store the `Lottery` class objects. Any other data structure used will result in point deductions.
3. Each ticket contains 6 digits (0-9). The digits can be repeated in the tickets. Examples include 123456, 003221, 529495, etc…
4. You will create a class called `Lottery` with the following required defined methods and attributes:
    a. An attribute of type String called `ticket` which holds the digit values. Make it private as that is good practice!
    b. Two Constructors
        i. One default constructor that sets `ticket` to an empty string.
        ii. One overloaded that takes a Random class reference.

c. GenerateRandomWinner

  i. Method that takes a Random class object reference as a parameter.

  ii. This method generates a random digits for the `ticket` and returns it. The type is String.

d. GenerateSelectWinner

  i. Method that takes a Random class object reference as a parameter and the max possible index of the array in a given scenario.

  ii. Method returns an index of the array of a winning ticket. (this is to test your two algorithms to see if it actually works by guaranteeing a winning `ticket` is found).

e. Solution1

  i. One of the solutions that finds the winning `ticket`.

  ii. This method should run O(n).

  iii. Returns a Boolean if found or not.

f. Soluton2

  i. One of the solutions that finds the winning ticket.

  ii. This method should run O(lgn).

  iii. Returns a Boolean if found or not.

g. You will need to create an accessor method to retrieve the ticket attribute.

h. You will need to create a sort method to sort the tickets.

i. You may create any additional methods that will assist in this programming assignment.

A runner file (`LotteryRunner.java`) has been provided for you to show you how the methods are called along with 5 test cases based on the size of the input array. Note, these test cases ran on the Eustis environment (if you run on your local system, the pseudorandom generator will produce different numbers which can potentially leading to a test case failing). One thing you will have to make sure is that your two algorithms satisfied the running time restrictions. These test cases ensure that your solution is correct in producing the desired output. <u>They do not test your run time.</u>

**What to submit:** Submit a file called `Lottery.java` to webcourses. You are not required to submit the runner file as that will be provided for the graders to test your code. Please make sure the runner file provided works for your code. Any name changes may cause your program not to work when graded, which will result in a lower score on the assignment and would not be changed.

**Important Note for running Eustis:** Many of you are probably using IDEs like Netbeans and Eclipse to build your Java Solutions. Please note you will need to separate the two files. In other words, do not put the runner file in the same package as `Lottery.java`. If you put both files in the same package, the command line will have issues with compiling. The workaround this is putting your `Lottery.java` file in a package called “`lottery`” and have the runner file import it (see the runner file for clarification). **Please make sure to name the package `lottery`.**