CSC 17c C++:Data Structures **Name:** Emiliano Manalo
Spring 2025
Midterm
04/21/25
Time Limit: Maximum of 7 days

The Midterm contains 4 pages and 12 questions. Total points possible are 200.

You should create a folder labeled Midterm that has folders labeled for each Problem when many files clearly are required in the problems solution. Zip all the work into 1 file which will be submitted on Canvas.

For Problems 1 to 3 – Provide Project Code, Spreadsheets, Plots, and written derivations. Note: The answers are well known. Stating the answer is not the solution. I want you to prove the answer in 3 different ways as we showed in class with an example sorting routine. A correct submission of all 3 types of analysis would rate an A for the problem, submission of 2 would rate a B for the problem and submitting only the timing analysis with curve fit data, plots, and analysis would rate a C.

---

1. (30 points) Analyze and compare the linear and binary search. See Github /Book /Midterm Folder. I have provided a project folder with the code.

   - Show O() by mathematical analysis – Show all work/algebraic summative derivations.
   - Show O() by operational analysis – Curve Fit with my program and tables/plots in Spreadsheet
   - Show O() by timing analysis – Curve Fit with my program and tables/plots in Spreadsheet

2. (30 points) Analyze, compare and contrast bubble sort with selection sort. See Github /Book /Midterm Folder. I have provided a project folder with the code.

   - Show O() by mathematical analysis – Show all work/algebraic summative derivations.
   - Show O() by operational analysis – Curve Fit with my program and tables/plots in Spreadsheet
   - Show O() by timing analysis – Curve Fit with my program and tables/plots in Spreadsheet

3. (30 points) Compare insertions, i.e. 1) push method with Simple Vector using arrays, 2) with Optimized Simple Vector using arrays, and 3) Simple Vector implemented with a Linked List. See Github/Book/Midterm Folder. I have provided a project folder with the code.

   - Show O() by mathematical analysis – Show all work/algebraic summative derivations.

- Show O() by operational analysis – Curve Fit with my program and tables/plots in Spreadsheet
- Show O() by timing analysis – Curve Fit with my program and tables/plots in Spreadsheet

4. (10 points) Complete the table, given the Order, Scale accordingly.

Big O Scaled								
Big O()	Second	Minute	Hour	Day	Month	Year	Decade	Century
$N^{1/3}$								
$N^{1/2}$								
N	128							
$Nlog(N)$								
N^2								
$N^2log(N)$								
2^N								
$N!$								

5. (10 points) Derive the order of the error with respect to the sin and cosine approximations.

Sin and Cos Error at x=1/N		
Series	sin(x)=x–x^3/3!+x^5/5!+...	cos(x)=1–x^2/2!+x^4/4!-...
Approx	$sin(x) \approx x$	$cos(x) \approx 1 - x^2/2$
Error Big O()	(sin series - approx) at x=1/N	(cos series - approx) at x=1/N

6. (15 points) Derive the O() for the Recursive vs. non-Recursive Fibonacci Function. See repository.

7. (15 points) Given 4 cards with 13 possible face values, calculate the probability of 1 pair, 2 pair, 3 of a kind and 4 of a kind. Simulate the results and compare to calculations.

8. (10 points) Given a biased coin analogy, if a bit vector is 40% full, what are the odds that 5 bits randomly chosen all fall within the filled section. Simulate the results and compare to calculations.

9. (10 points) Recursive function. Provide code and test sufficiently. Create 2 types of Power functions (Hint for O (log(n)) split into even and odd conditions.

$$\forall (y, x) \in \mathbb{R}, \forall n \in \mathbb{N}_0 : (y(n) = O(n) \ \cap \ y(n) = O(log(n)))$$
$$y = x^n$$

10. (10 points) Recursive function. Provide code and test sufficiently for the conditions displayed:
$$\forall x \in \mathbb{R} \ \cap \ -1 \leq x \leq 1 \ \cap \ \delta x = 0.1$$

$$g(2x) = \frac{2g(x)}{1 + g(x)^2}$$

with base conditions of

$$|x| < \epsilon \ \cap \ \epsilon = 10^{-6} \ \rightarrow \ g(x) = x - x^3/6$$

11. (10 points) Mutual Recursive functions. Provide code and test sufficiently. Recursive function. Provide code and test sufficiently for the conditions displayed:

$$\forall x \in \mathbb{R} \ \cap \ |x| \leq arctan(1) \ \cap \ \delta x = 0.1$$

$$C(2x) = \frac{C(x)S(x)}{2} \ \cap \ S(2x) = \frac{C^2(x)S^2(x)}{C^2(x) - S^2(x)}$$

with base conditions of

$$|x| < \epsilon \ \cap \ \epsilon < 10^{-6} \ \rightarrow \ C(x) = 1/x + x/6 \ \cap \ S(x) = 1 + x^2/2$$

12. (20 points) Code the mode problem first with the Set container, then use the Map container from the STL. I have already coded the problem for you without those containers. You are to take my code and reduce the number of lines required for the mode function by showing your expertise of Sets first then using Maps second. The idea is that Sets should reduce the code required and Maps should reduce it even further. Implement the function in the least amount of code for each container.

1. Linear and Binary Search
   Math Analysis -10
   Operator Analysis -10
   Time Analysis -10

2. Bubble and Selection Sort
   Math Analysis -10
   Operator Analysis -10
   Time Analysis -10

3. Simple Vector Comparison
   Math Analysis -10
   Operator Analysis -10
   Time Analysis -10

4. Big O Scaled Table -10

5. Order of the Error -10

6. Fibonacci Recursive vs. Non-Recursive
   Math Analysis -10
   Operator Analysis -5

7. Cards Hands
   Sim -5
   Analysis -10

8. Bit Vector
   Sim -5
   Analysis -10

9. Power Recursion
   O(n) -5
   O(log(n)) -5

10. Recursive Function -10

11. Mutual Recursion -10

12. Mode reduction in coding lines
    Sets -10
    Maps -10