Exam 1:

Due Date: **Sunday, October 11, at 11:59pm.**

This exam contains three problems asking multiple questions. Please answer each question in detail with clear explanation. :)

Problem 1.

A. What is the growth of the below functions? **Explain in detail and show ALL the work**.

```
Test1(a)
    n = a.length;
    key = bad_fun($n^4$); // time complexity of bad_fun(k) is O(k!)
    print(key);
    binary_search(a,key);
end
```

```
Test2(n)
    for i = 1:n
        if(i<n/10)
          binary_search(a, key); //a.length = $n^3$, key does not exist in a
        else
          linear_search(a, a[1]); //a.length = $n^3$, key == a[1]
        end
    end
end
```

B. Compare the growth of $T_{Test1}(n)$ and $T_{Test2}(n)$. **Show all the work.**

C. Let's say you can finish running $T_{Test2}(10^6)$ in *1* sec. Could you estimate when you finish running $T_{Test1}(100)$?

Problem 2.

A sorted array and a random number are given to you. Develop an algorithm to find the total number of the repetitions of the given number.

Example 1: Input: a = [0, 1, 1, 2, 2, 2, 3, 3, 6], key = 2 ➔ Output: "2 was repeated 3 times."

Example 2: Input: a = [0, 0, 2, 2, 3, 9, 10, 12, 15], key = 10 ➔ Output: "10 was repeated 1 times."

Example 2: Input: a = [0, 1, 3, 8, 12], key = 5 ➔ Output: "5 was repeated 0 times."

A. How would you find the total number of repetitions for the given number? (**Note**: If you have multiple answers in mind, break them apart and explain each one separately.) Explain each solution/algorithm in a few lines.

B. Write the pseudocode for the best algorithm you came up with.

C. Implement your answer using any programming language you want to.

D. What is the time complexity of your answer? **Explain in detail and show all the work.** (**Note**: If possible, break your code/pseudocode to different parts, calculate the runtime for each step and then try to calculate the total running time based on that.)

Problem 3. A random array of size *n* is given to you. You know that the elements in the array are nonnegative integers less than *n*. Develop an algorithm to find the mode (the value that appears most) and the numbers repeated more than once.

Example 1: *Input*: a = [6, 0 ,1, 5, 1, 1, 4, 5], ➔ *Output1*: "1 is the mode."
Output2: "5 was repeated 2 times,
1 was repeated 3 times."

Example 2: *Input*: a = [0, 2, 4, 2, 2, 0, 0, 5, 4], ➔ *Output1*: "0 and 2 are the mode."
Output2: "0 was repeated 3 times,
2 was repeated 3 times,
4 was repeated 2 times.""

(**Hint**: Remember the algorithms we learned in the class (searchings and sortings). Could you pick a good one and use parts of it to solve this question?)

A. How would you find the mode and the numbers occurring more than once? (**Note**: If you have multiple answers in mind, break them apart and explain each one separately.) Explain each solution/algorithm in a few lines.

B. Write the pseudocode for the best algorithm you came up with.

C. Implement your answer using any programming language you want to.

D. What is the time complexity of your answer? **Explain in detail and show all the work**. (**Note**: If possible, break your code/pseudocode to different parts, calculate the runtime for each step and then try to calculate the total running time based on that.)