Exam 2:

Due Date: **Monday, April 13, at 6:00pm.**

This exam contains two problems, each asking _four_ questions. Please answer each question in detail with clear explanation. :)

· ·

Problem 1. Find the index of the smallest number in a sorted array where the first k numbers were shifted to the end.

An example of a sorted array where the first k numbers were shifted to the end:

| 1, 2, 3, 4, 5, 6, 7 | | 3, 4, 5, 6, 7, 1, 2 |
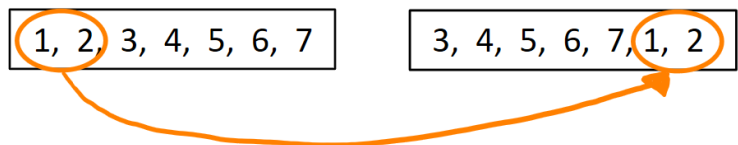
Example 1: _Input_: a = [2, 5, 8, 10, 12, 0, 1] ➜ _Output_: 5

Example 2: _Input_: a = [1, 6, 9, 10] ➜ _Output_: 0

Example 3: _Input_: a = [20, 30, 40, 1, 5, 10] ➜ _Output_: 3

A. How would you find the index of the smallest 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 2. You are given k sorted arrays in descending order of size n. Develop an algorithm to merge them into a single sorted array of size kn.
(**Hint 1**:
      i. Can you find the largest number in each array? (This step will give you **k** numbers)
      ii. How would you find the max value among these **k** numbers? What is the best algorithm you can come up with?
      iii. Do you know from which array the max value came from?
      iv. Now, how would you find the 2nd largest element?

Hint 2: "The root has the largest value in a <u>max heap</u>". Can you guess where knowing this statement might help you?

Example:
a1 = [8, 4, 2, 0],
a2 = [20, 15, 5, 3],
a3 = [10, 7, 6, 1],
Output: Merged array: [20 15 10 8 7 6 5 3 2 1 0]

A. How would you merge these k arrays into a single sorted array? (**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.)